SDC Data Provider Cheat Sheet

Table of Contents

[1 Software Prerequisites 2](#_Toc4503173)

[2 Credential update scripts 2](#_Toc4503174)

[3 Upload Procedures 3](#_Toc4503175)

[3.1 S3 ingest bucket uploads 3](#_Toc4503176)

[3.2 Kinesis Firehose Real Time Ingestion 4](#_Toc4503177)

[4 Validating upload 4](#_Toc4503178)

# Software Prerequisites

In order to upload data to the SDC, install the following software products and any of their software requirements onto the machines that will be sending data. Website links for installation instructions are provided where applicable.

* Python version 3.5 or later
  + Windows: <https://www.python.org/downloads/windows/>
  + Mac: <https://www.python.org/downloads/mac-osx/>
  + Linux:
    - <https://www.python.org/downloads/source/> or
    - sudo apt-get install python3
* Amazon AWS Command Line Interface. Choose the correct version for the platform you are running. <https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html>
* Install Python jq library   
  <https://stedolan.github.io/jq/download/>
* Install Git Bash   
   <https://git-scm.com/downloads>

# Credential update scripts

In order to increase security within the SDC and prevent malicious attempts to extract or upload data, provider credentials are set to time out after 4 hours. Providers should update their credentials before starting a data upload. Credentials can be updated using the scripts included in the welcome email. There should be a script included for Linux and Windows operating systems.

In addition to the attached scripts, the welcome email will also include a document with a set of values that need to be updated in the credential scripts prior to running. Please make sure to update the values listed below in the credential script so the script will retrieve the correct credentials.

* API\_KEY
* AUTH\_CODE
* ROLE\_NAME
* API\_END\_POINT

Please ensure the following folder and files exists under home directory.

     .aws

        ----config

       ----credentials

Run the downloaded shell script which will generate a temporary access/secret keys and update ~/.aws/credentials files by creating a new profile named **sdc-token.** if you already have the profile named **sdc-token**, it will overwrite by updating the credentials.

**Note: -** Windows users needs to open Git Bash terminal, navigate to the sh script folder and run the sh script (i.e. ./dot-sdc-refresh-token.sh)

If your uploads are very large and may take longer than the credential expiration time, please send an email with a description of the issue to the [support@securedatacommons.com](mailto:support@securedatacommons.com) so we can work with you to come up with a specific solution to address your needs.

# Upload Procedures

There are currently two options for methods of uploading data to the SDC. For batch uploads on an hourly, daily, monthly, or other basis, S3 uploads should be used. For near real-time streaming of data to the SDC, Firehose ingest streams should be used.

## S3 ingest bucket uploads

Uploads to your provider-specific S3 ingest bucket can be accomplished directly from a command line in either the Windows Command Prompt, or a Linux or Mac-OSX Terminal. More detailed documentation for usage of the AWS CLI to upload files to S3 buckets can be found here: <https://docs.aws.amazon.com/cli/latest/reference/s3/cp.html>.

For data upload purposes, the most useful commands will be copy commands from a local source to the remote, AWS s3 ingest bucket. Use the command below to upload a single object to S3:

>aws s3 cp <local object filename or filepath> s3://<ingest bucket name>/<data type name>/ --profile sdc-token

Where “ingest bucket name” is the provider specific ingest bucket given in the welcome email. The “data type name” shown in the command should be a short, descriptive term describing what is the data type being uploaded. This value should have the following properties:

* be unique for each type of data being uploaded,
* consist of all UPPERCASE letters,
* should not contain spaces, but may contain underscores or dashes, and
* may consist of subfolders if that is applicable to the specific data type.

If you would like assistance or further guidance choosing a descriptive label, send an email to [support@securedatacommons.com](mailto:support@securedatacommons.com).

Uploading multiple files at a time recursively uses a similar command:

>aws s3 cp /<local folder path>/ s3://<ingest bucket name>/<data type name>/ --recursive --profile sdc-token

In this case, the “local folder path” should be a folder path with the entire set of objects to be uploaded to the SDC. The “--recursive” tag indicates to upload all the files in that folder.

**As an additional cautionary note, any subfolder structure that exists in the folder described by “local folder path” will be copied over to the S3 Ingest bucket. This is NOT ALLOWED in the SDC. Please make sure there are only single files in the local folder when performing both single uploads and recursive, multiple-object uploads.**

Ingesting data into the S3 ingest buckets can also be accomplished easily using batch scripts on Windows or shell scripts in OSX or Linux. There are also python libraries that interface with AWS, so there are multiple avenues by which to perform ad-hoc, batch uploads of data to the SDC.

## Kinesis Firehose Real Time Ingestion

Uploads through real time ingestion infrastructure can also be done using the AWS CLI. For full documentation for Kinesis data streams, see the AWS site here: [https://docs.aws.amazon.com/cli/latest/reference/kinesis/index.html#](https://docs.aws.amazon.com/cli/latest/reference/kinesis/index.html).

The most useful command will be “put-record.” See the documentation on the “put-record” command in the web pages linked above for instructions on the use of this command to upload records in real time.

As with S3, Kinesis Firehose can also interface with batch script files as well as python and other scripting or programming languages. For more information or help setting up a real time ingest stream, send an email to [support@securedatacommons.com](mailto:support@securedatacommons.com).

# Validating upload

For security reasons and to avoid the possibility of tampering with uploaded data, data files that are ingested through S3 buckets are moved immediately to a different location when uploaded. Upon successful upload of data files to the ingest bucket using the above step, data will be moved to the raw submissions bucket or “Data Lake.” Background processes will move the data from ingest bucket to the raw submissions buckets under a folder labelled based on the date it was uploaded. As a data provider, you will have a folder in the data lake that contains all of the data you upload.

Local Object 🡪🡪 Ingest Bucket 🡪🡪 Raw Submissions Bucket

Data uploads can be verified by running the below AWS CLI command on the raw submissions bucket to list the objects there. The raw submissions bucket name is provided in the table below the command. The “project name” and “data provider name” were provided in the welcome email.

**AWS CLI Command: -**

>aws s3 ls s3://<raw submissions bucket name>/<project name>/<data provider name>/<data type name>/ --profile sdc-token

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Dev** | **Test** | **Prod** |
| **Raw Submissions Bucket Name** | dev-dot-sdc-raw-submissions-911061262852-us-east-1 | test-dot-sdc-raw-submissions-911061262852-us-east-1 | prod-dot-sdc-raw-submissions-911061262852-us-east-1 |