International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V7I1P113
Eureka Vision Publication | Volume 7, Issue 1, 97-100, 2026

Original Article

Efficient Bulk File Ingestion into Data Lake using SMB and FTP protocols

Vamshi Krishna Malthummeda
Independent Researcher, USA.

Received On: 15/12/2025 Revised On: 16/01/2026 Accepted On: 24/01/2026 Published On: 04/02/2026

Abstract - It is very essential for companies in the retail industry to exchange sensitive information like Purchase Orders, Invoices, Payments Information, Contracts and other types of information for improving efficiency, reducing costs, and for strengthening business relationships. The sensitive information also gets exchanged between different teams within an organization for better collaboration, efficiency and to extract valuable insights. This paper introduces a cost effective and secure approach which is tailored for bulk ingestion into databricks data lake using SFTP and SMB protocols. The proposed file transfer framework makes use of python packages like paramiko (which implements the Secure Shell SSHv2 protocol for secure file transfer over unsecured network) and pysmb(which implements SMB protocol for secure file sharing across different operating systems within a network) to transfer the files into the databricks volume for further processing by transformation pipelines in databricks cluster. Using this methodology various types of file transfers like synchronous small file transfer, asynchronous large file transfer and continuous file transfer can be achieved. Deletion of files on the remote server after a configured number of successful transfers is also achieved using the proposed methodology. The key findings of this implementation are as following: simplified setup procedure, less development effort and decent performance. The findings suggest that the proposed framework with data integrity and remote file management capabilities is a secure, simplified, versatile and a reliable choice for transferring sensitive files.

Keywords - SFTP, SMB Protocol, SSH, Databricks, Data Lake, Pyspark, Python.

1. Introduction

In retail industry, businesses give monetary incentives to partners who meet an agreed upon sales target and inspire deeper loyalty and drive desired channel purchasing behavior. Rebates, which are financial incentives offered by businesses to encourage purchases and loyalty, can significantly impact a business's financial performance, partner relationships, and overall operational efficiency [1]. Businesses will come to know that partners met the sales targets through the transaction records which will be received through a secured channel. The account details of the partner organizations and their point of contacts are maintained on the internal servers with different operating systems.

2. Secure File Transfer Protocol(SFTP)

The partners/customers send transaction information to the business using SSH File Transfer Protocol also known as Secure File Transfer Protocol(SFTP). SFTP is ideal for transferring sensitive or confidential information like transactions. Before we discuss SFTP, let’s quickly review the File Transfer Protocol(FTP). FTP is a protocol for transferring data between a computer and remote computer/server over an internet connection. FTP has 2 communication channels.

- Control Channel: FTP makes use of a control connection for sending information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc. The control channel connection is initiated on port number 21.
- Data Channel: For sending the actual file, FTP makes use of a data connection. A data connection is initiated on port number 20.

SFTP also uses a client server connection to facilitate file transfer, but SFTP file transfers are done over the control channel and there is no need to open a separate data channel to complete file transfer. SFTP provides robust security compared to FTP by using SSH (Secure Shell) to encrypt both the data and the authentication information. This ensures that all information transferred between the client and server is encrypted, making it much harder for unauthorized parties to intercept and read the data. SFTP also supports public key authentication, adding an extra layer of security by verifying the identity of the server and client [[2]]. SFTP file transfer happens over an SSH session on TCP port 22.

3. Server Message Block Protocol(SMB)

The account details and point of contact details of the partner organizations are shared between the teams by placing them on the internal network servers and the teams which need those files access them using SMB protocol.

The SMB protocol facilitates file and printer sharing across multiple operating systems. SMB's primary function is to facilitate file sharing within a network. This streamlines teamwork and collaboration, as users can easily access, read, edit, and save files from different devices.

The SMB protocol works on a client (Initiates connection and requests files or services)-server(Listens for the requests and delivers the resources or services) interaction model[[3]].

Below are the steps involved in the SMB Request-Response cycle:

- Session Establishment: Client initiates a session with the server over the TCP port:445(for the direct SMB over TCP/IP connections)
- Authentication: Client submits authentication credentials and server verifies credentials and grants access if the credentials are valid.
- Resource Access: Client requests access to a file and server checks permission and grants access if authorized.
- Data Transfer: Client & server exchange commands to read, write or modify the resource. Data is transferred using SMB-encapsulated network packets.
- Session Termination: Client sends the command to close the session, and server acknowledges ending the session.

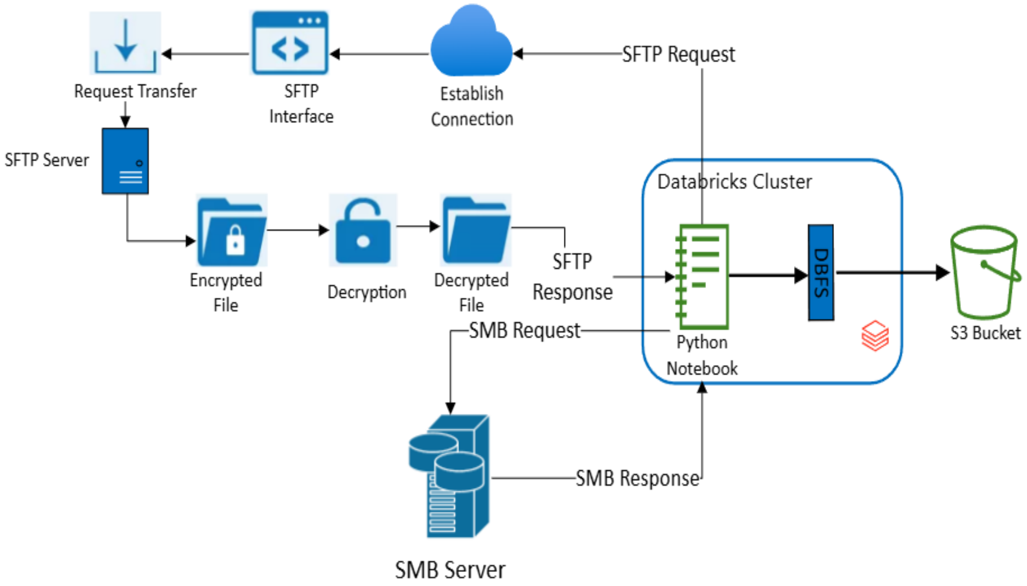

Figure 1. File Transfers from SFTP and SMB Servers into Databricks Cluster

The transaction files from various partners & payment systems received via SFTP are joined with the customer/partner account information imported from the SMB internal servers into the databricks data lake. Imported transactions and accounts data is subjected to various transformations to identify the customers who met the targets set. Based on the types of targets reached, appropriate incentives in the form of rebates & discounts are provided to the customers/partners.

4. Solution Description

The framework for importing the files from SFTP and SMB servers and ingesting those files into data lake is developed in databricks. The databricks job hosts the client process which establishes the connection to the SFTP and SMB servers. The client process makes use of paramiko[[5]] python package which implements the SSHv2 protocol to create a helper class for downloading the files from SFTP server into the databricks volume.

The SFTPDownloader is implemented as a Python context manager object[[4]]. The context managers are primarily used for managing resources such as files, network connections, database connections or locks. They guarantee that the resource is properly setup before the code block executes and cleaned up afterward regardless of whether the block completed successfully or an exception is raised.

The SFTPDownloader class needs 3 important dunder (double underscore) functions to be defined:

- __init__ function: which accepts the SSH connection parameters like host, port(default 22), username and password
- __enter__ function: calls another instance function connect() which establishes connection to the SFTP server using paramiko package utility functions(takes care of setup & resource acquisition related tasks when SFTPDownloader is used as context manager)
- __exit__ function: which closes the established connection(takes care of cleanup activities after code block execution is completed)

SFTPDownloader also implements the following functions:

- connect: to establish SFTP server connection
- listdir: lists the files on SFTP server path
- mkdir: creates directory on SFTP server at the specified path

- put_from_path: to copy local files to a specified path on remote SFTP server
- put: to copy files from databricks volume to specified path on remote SFTP server
- get_to_path: Copy remote files from SFTP server to designated local file location
- get: Copy remote files from SFTP server to designated volume location on databricks with additional configurations like splitting the large file into small chunks and copy it asynchronously
- remove: to remove the file from remote SFTP server
- rmdir: to remove the directory from remote SFTP server
- rename: to rename the file on SFTP server
- stat_size: get the file size
- listdir_attr: returns the list of file attributes for the specified directory path

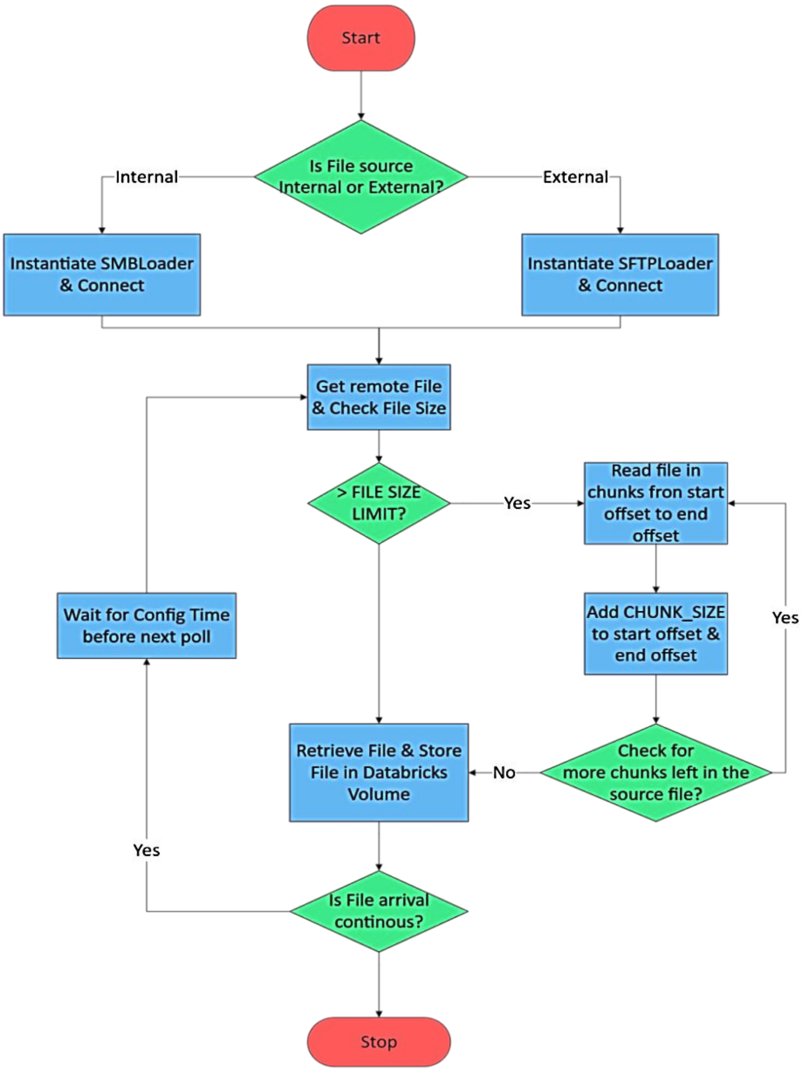
Start
Is File source Internal or External?
Internal
External
Instantiate SMBLoader & Connect
Instantiate SFTPLoader & Connect
Get remote File & Check File Size
> FILE SIZE LIMIT?
Yes
Read file in chunks fron start offset to end offset
Add CHUNK_SIZE to start offset & end offset
Yes
Wait for Config Time before next poll
Retrieve File & Store File in Databricks Volume
No
Check for more chunks left in the source file?
Yes
Is File arrival continous?
Stop

Figure 2. File Download from SFTP and SMB Servers Flow

Similarly, SMBDownloader is also implemented as context manager object which manages the connection to SMB server using pysmb package. These downloaders will poll the remote servers for new files arrival and transfer them to the databricks cluster. Large files on the remote SFTP and SMB servers are divided into parts and transferred asynchronously. The data downloaded using SFTPDownloader and SMBDownloader is directly uploaded into databricks volume using with open() file handler in the databricks cluster client process.

5. Conclusion

Sharing data between organizations in a secure way to prevent unauthorized access, theft, protecting privacy, maintaining trust and preventing financial & reputational damage is of paramount importance for the organizations. The solution discussed above to address these issues, downloads the external files using SFTP and internal files using SMB into the databricks volumes followed by data transformations is the simplest, secured and straight forward approach. Employing SFTP for batch, non-real time and bulk file transfer over the internet. Using SMB for transferring data within the network and using databricks environment for the data transformations provides all the stakeholders involved the desired results of security, data integrity and extraction of valuable insights with minimal investment. SFTP & SMB protocols have their own limitations like weak passwords, lack of multi-factor authentication etc. which need to be addressed to protect the sensitive information and fill the security gaps.

References

[1] Shivaraj, G. (2024). OPTIMIZING REBATE MANAGEMENT IN SUPPLY CHAIN OPERATIONS. *Technology (IJARET)*, *15*(3), 110-118.

[2] Bomma, H. P. (2021). Navigating the Challenges of Data Encryption and Compliance Regulations: FTP vs. SFTP. *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, *9*(5), 1-6.

[3] Ts, J., Eckstein, R., & Collier-Brown, D. (2003). *Using Samba*. " O'Reilly Media, Inc.".

[4] Behler, J. A. C. (2023). *Assessing Python Bindings of C Libraries with Respect to Python Idiomatic Conformance* (Master's thesis, Kent State University).

[5] Welcome to Paramiko's documentation! Paramiko documentation.