

**3. Data Ingress Patterns**

Pull Methods  
API Calls: Scheduled or on-demand retrieval from external/internal APIs.  
Scheduled DB Calls: Periodic extraction from databases.  
Scheduled SCP/SFTP Calls: Automated file transfers from remote servers.  
SharePoint/Confluence: Manual or automated document ingestion.  
JDBC/ODBC Calls: Direct database connectivity for data extraction.

I am working on creating a doc for each of the methods above on aws ingestion,  
Need to elaborate on each, different ways , pros , cons and limitations, cost considerations

Below is an elaboration on each data ingress pattern for AWS, including **methods**, **pros/cons**, **limitations**, and **cost considerations**. This is structured for easy inclusion in your documentation.

**1. API Calls (Scheduled or On-Demand Retrieval)**

**Methods:**

* Use AWS Lambda with Amazon API Gateway to pull data from external/internal APIs.
* Schedule Lambda invocations using Amazon EventBridge (CloudWatch Events) for periodic pulls.
* Direct integration with AWS Glue or custom EC2 scripts for more complex workflows.

**Pros:**

* Near real-time or on-demand access to fresh data.
* Highly flexible and scalable.
* Supports both synchronous and asynchronous patterns (e.g., polling for status, then retrieving data)[[1]](#fn1)[[2]](#fn2).

**Cons:**

* API rate limits may restrict throughput.
* Error handling and retries must be managed.
* Can be complex if authentication, pagination, or data transformation is required.

**Limitations:**

* Dependent on external API availability and reliability.
* Latency can be introduced by API response times.
* Large data volumes may require batching or chunking[[2]](#fn2).

**Cost Considerations:**

* **API Gateway**: Pay-per-request model (e.g., $3.50 per million REST API calls in US East), plus data transfer fees[[3]](#fn3)[[4]](#fn4)[[5]](#fn5).
* **Lambda**: Pay for compute time and number of invocations.
* **Additional**: Caching increases cost but can reduce API calls[[5]](#fn5).

**2. Scheduled Database Calls**

**Methods:**

* Use AWS Glue, Lambda, or Data Pipeline to schedule queries against RDS, Aurora, Redshift, or external databases via JDBC/ODBC.
* Extract results and store in S3 for further processing.

**Pros:**

* Automates extraction from structured sources.
* Integrates well with AWS analytics and ETL services.
* Can be tuned for off-peak hours to minimize impact on source systems.

**Cons:**

* May require VPN or VPC peering for on-premises databases.
* Schema changes in source DB can break ingestion jobs.
* Not real-time; data freshness depends on schedule.

**Limitations:**

* Limited by query performance and database throughput.
* Large extracts may require partitioning or incremental loading.

**Cost Considerations:**

* **Glue**: Charged per Data Processing Unit (DPU) hour.
* **Lambda**: Pay per invocation and compute time.
* **RDS/Redshift**: Data transfer and query costs.

**3. Scheduled SCP/SFTP Calls**

**Methods:**

* Use AWS Transfer Family (SFTP/SCP) to receive files directly into S3.
* Lambda or Glue jobs can be triggered upon file arrival for downstream processing.
* For outgoing pulls, use EC2, Lambda, or Glue with custom scripts to fetch files from remote servers[[6]](#fn6).

**Pros:**

* Secure, encrypted file transfer.
* Compatible with legacy systems and batch data feeds.
* AWS Transfer Family is fully managed and integrates with S3.

**Cons:**

* Not real-time; data is only as fresh as the file delivery schedule.
* File format and schema consistency must be managed.
* Requires handling of partial/incomplete file transfers.

**Limitations:**

* Throughput limited by network and file size.
* Manual intervention needed for failed transfers unless automated retries are implemented.

**Cost Considerations:**

* **AWS Transfer Family**: Charged per hour of endpoint usage and per GB of data transferred.
* **S3 Storage**: Standard S3 storage and request costs apply.
* **EC2/Lambda**: Additional compute costs for custom scripts.

**4. SharePoint/Confluence (Manual or Automated Document Ingestion)**

**Methods:**

* Manual: Download documents and upload to S3 via AWS Console or CLI.
* Automated: Use custom scripts or third-party connectors (e.g., AWS Lambda with Microsoft Graph API for SharePoint) to periodically sync documents.

**Pros:**

* Centralizes unstructured documents in AWS for further processing or search.
* Automation reduces manual effort and errors.

**Cons:**

* APIs for SharePoint/Confluence may have rate limits and require complex authentication.
* Metadata and permissions mapping can be challenging.

**Limitations:**

* Not natively supported by AWS; requires custom or third-party solutions.
* Potential for data duplication or missed updates if sync logic is not robust.

**Cost Considerations:**

* **Lambda/EC2**: Compute costs for automation.
* **API Gateway**: If used for orchestration.
* **S3**: Storage and request costs.

**5. JDBC/ODBC Calls (Direct Database Connectivity)**

**Methods:**

* Use AWS Glue, Data Pipeline, or custom EC2/Lambda jobs to connect to databases using JDBC/ODBC drivers.
* Extract data and write to S3, Redshift, or other AWS services.

**Pros:**

* Direct access to relational data.
* Supports incremental loads (e.g., using timestamps or change data capture).

**Cons:**

* Requires secure network connectivity (VPC, VPN, Direct Connect).
* Driver compatibility and versioning can be an issue.
* Not suitable for high-frequency or real-time ingestion.

**Limitations:**

* Performance limited by source DB and network.
* May impact source database performance during extraction.

**Cost Considerations:**

* **Glue**: Per DPU hour.
* **EC2**: Instance and data transfer costs.
* **Redshift Spectrum**: If querying directly from S3.

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Pros | Cons/Limitations | Cost Considerations |
| API Calls | Real-time, flexible, scalable | Rate limits, error handling, API costs | API Gateway, Lambda, data transfer fees |
| Scheduled DB Calls | Automated, structured data | Not real-time, schema drift, connectivity | Glue/Lambda, RDS/Redshift, transfer costs |
| SCP/SFTP | Secure, legacy compatibility | Not real-time, file management, network limits | Transfer Family, S3, compute costs |
| SharePoint/Confluence | Centralizes documents, automation | API complexity, metadata mapping | Lambda/EC2, S3, possible API Gateway |
| JDBC/ODBC | Direct relational access, incremental | Network/security, driver issues, performance | Glue, EC2, Redshift, data transfer |

For each method, AWS provides a pay-as-you-go model, so costs scale with usage. Free tiers are available for some services (e.g., API Gateway, Lambda), but large-scale or frequent ingestion can lead to significant charges[[3]](#fn3)[[4]](#fn4)[[5]](#fn5)[[7]](#fn7). Always estimate based on projected usage and monitor with AWS Cost Explorer.

⁂

1. <https://github.com/linkedin/data-integration-library/blob/master/docs/patterns/summary.md>

1. <https://developer.salesforce.com/docs/data/data-cloud-ref/guide/c360a-api-salesforce-cdp-ingestion.htm&rut=3918ab8dc8e41bf36b6195456724fe8bd9919ae3480f1efddee7734ce3c7c329>

1. <https://ably.com/topic/amazon-api-gateway-pricing>

1. <https://aws.amazon.com/api-gateway/pricing/>

1. <https://dataengineeracademy.com/blog/aws-api-gateway-pricing-factors-and-cost-structure/>

1. <https://www.educative.io/courses/data-engineering-foundations/ingestion-methods-sftp-api-and-object-storage>

1. <https://aws.amazon.com/servicecatalog/pricing/>