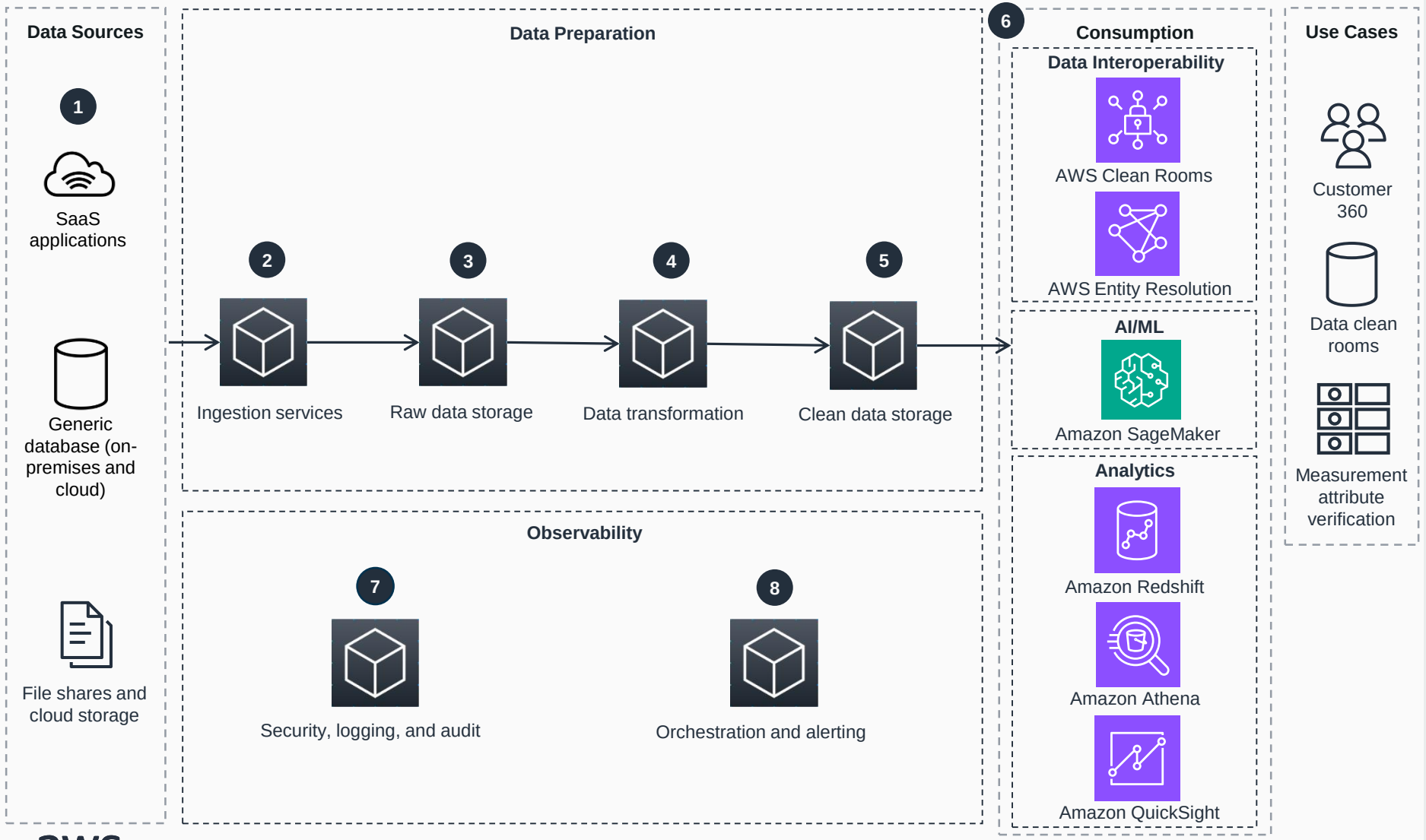
Guidance for Connecting Data Sources for Advertising and Marketing Analytical Workloads on AWS

This architecture diagram shows an overview of how to connect data sources stored in a variety of data sources to AWS.



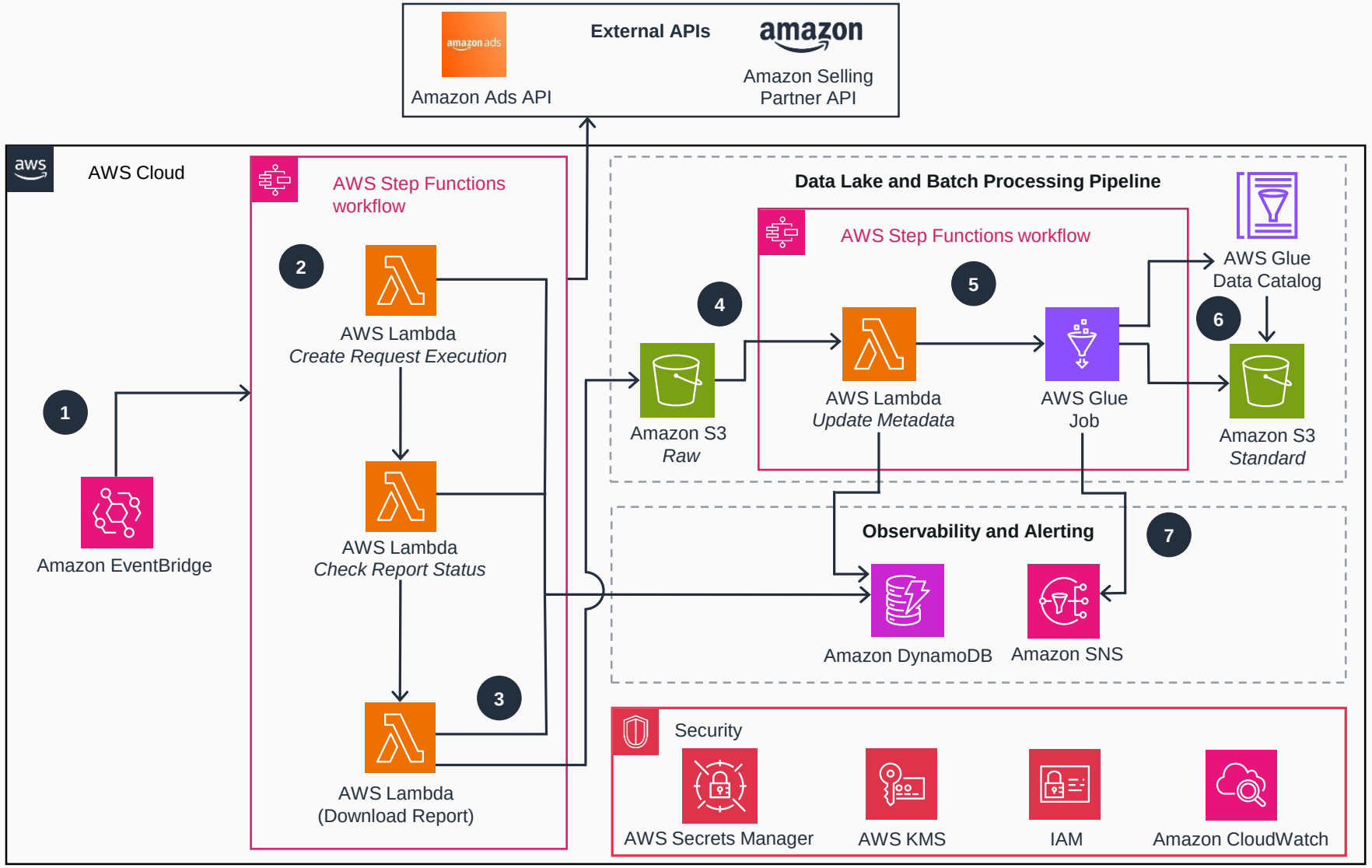
1. Sources of data needed for advertising and marketing analytics belong to one of the three categories: software as a service (SaaS) applications, relational databases, or file storage.
2. Use AWS services purpose-built for data ingestion to connect and pull data from data sources. The subsequent architecture patterns detail the ingestion service for each type of data source.
3. Use a cloud data storage “raw” zone as the destination for data ingestion services.
4. Use extract, transform, load (ETL) data processing jobs to transform data in a way that meets data consumption needs.
5. Store the transformed data in a cloud data storage “clean” zone. Catalog the data as relational tables in a data catalog service.
6. To build analytical applications, make the cataloged data available to consuming services such as **AWS Clean Rooms**, **AWS Entity Resolution**, **Amazon SageMaker**, **Amazon Redshift**, **Amazon Athena**, and **Amazon QuickSight**.
7. Build a unified observability stack that delivers the following functionality: a workflow metadata repository; workflow trigger events; chain tasks together to form an end-to-end workflow with consumption workloads; ability to generate observability notifications; and log capture and detailed observability dashboards.
8. Implement security and access control to achieve the following functionality: least privilege access to specific resources and operations; encryption for data at rest and data in transit; storage of hashing keys for personally identifiable data (PII) data; and monitoring of logs and metrics across all services used in this Guidance.

Reviewed for technical accuracy February 19, 2024

AWS Reference Architecture

Connecting Amazon Ads and Amazon Selling Partner Data to AWS – API Pull Pattern with AWS Lambda

This architecture diagram shows data ingestion and integration patterns for the Amazon Ads and Amazon Selling Partner APIs.



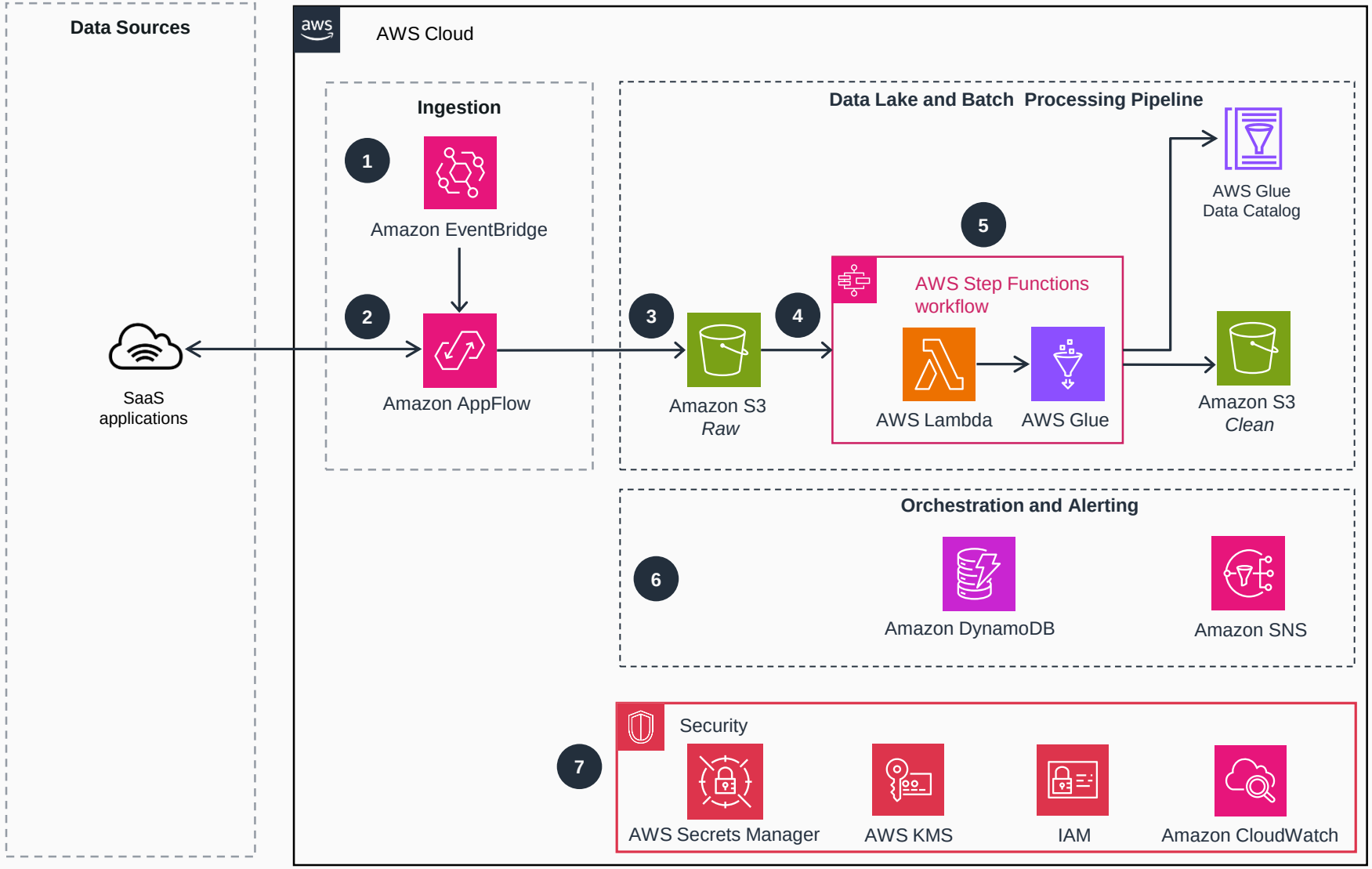
1. **Amazon EventBridge** schedules a job that starts an **AWS Step Functions** state machine. The state machine processes a series of **AWS Lambda** functions to facilitate report creation.
2. The state machine invokes a **Lambda** function (*Create Request Execution*) to create a report request from the Amazon Ads API or Amazon Selling Partner API.
3. Store API credentials within **AWS Secrets Manager**, and use them when making calls to the APIs. The state machine then moves into a series of polling steps, invoking a **Lambda** function (*Check Report Status*) to check the report request status before downloading the report.
4. **Amazon DynamoDB** stores the metadata for each downloaded report.
5. The **Lambda** function (*Download Report*) writes the report into a raw **Amazon Simple Storage Service (Amazon S3)** bucket with a prefix that contains the specific report type and report date. **Lambda** uses the Amazon-managed **AWS Key Management Service (AWS KMS)** key to encrypt the reports as they're written to the **S3** bucket.
6. A **Step Functions** state machine is invoked by notifications from **S3** objects as they're inserted into the bucket. When no more objects are received after a given time, the data transformation **Step Functions** state machine starts and invokes a **Lambda** function.
7. A **Lambda** function (*Update Metadata*) stores the task token for the **Step Functions** execution ID in the **DynamoDB** table. An **AWS Glue** job processes the data and reads the data from the raw **S3** bucket and transforms it to a usable format.

Reviewed for technical accuracy February 19, 2024

AWS Reference Architecture

Connecting SaaS Application Data to AWS – API Pull Pattern with Amazon AppFlow

This architecture diagram shows introduces data ingestion and a pull pattern for data available in SaaS applications.



1. **EventBridge** schedules a job that starts the **Step Functions** state machine, which includes the launch of **Amazon Appflow**.

2. The **Amazon AppFlow** flow starts by opening the connection to the external data provider and requesting data. The external provider responds with data to **Amazon AppFlow**.

3. **Amazon AppFlow** puts the data into the raw **S3** bucket and specified prefix. **Amazon AppFlow** uses an **AWS KMS** key to encrypt objects written to the raw **S3** bucket.

4. A **Step Functions** state machine is initiated by notifications from **Amazon S3** as the Guidance stores objects in the bucket. When no more objects are received after a given time, the data transformation **Step Functions** starts and continues with the common flow.

5. A **Lambda** function stores the task token for the **Step Functions** execution ID in the **DynamoDB** table and invokes the **AWS Glue** job. The **AWS Glue** job runs, reads data from the raw bucket, and transforms it. **AWS Glue** writes the transformed data to the clean **S3** bucket. **AWS Glue Data Catalog** metadata is also written out. The **AWS KMS** customer managed key (CMK) created by this stack encrypts the bucket contents and the **AWS Glue** metadata.

6. By storing the **Step Functions** workflow metadata and execution information, **DynamoDB** builds unified observability. **Amazon Simple Notification Service (Amazon SNS)** then generates observability notifications.

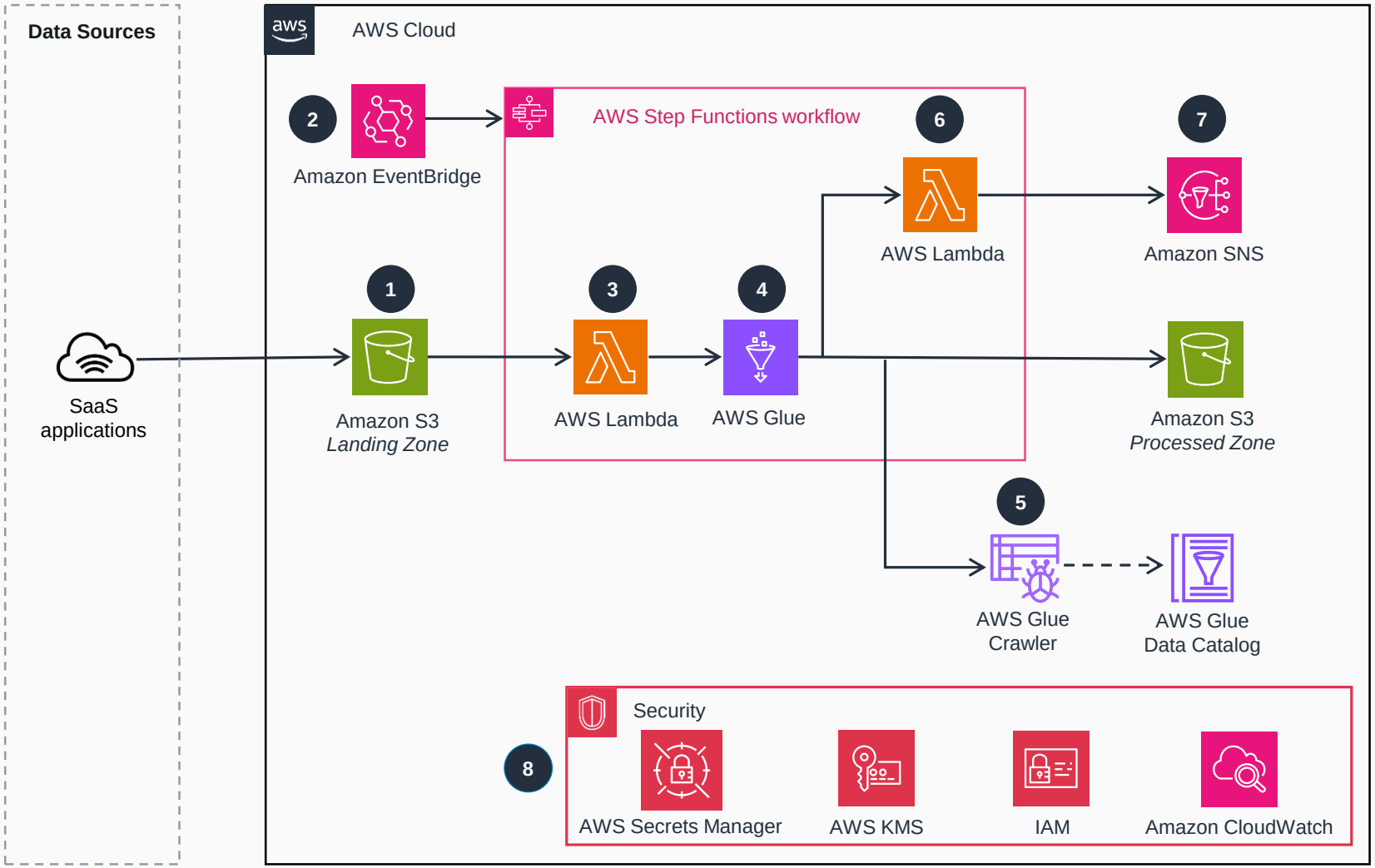
7. Use the following AWS services for security and access: **AWS Identity and Access Management (IAM)** enables least privilege access to specific resources and operations. **AWS KMS** provides encryption for data at rest and data in transit. **Secrets Manager** provides hashing keys for PII data. **Amazon CloudWatch** monitors logs and metrics across all services used in this Guidance.

Reviewed for technical accuracy February 19, 2024

AWS Reference Architecture

Connecting SaaS Applications to AWS – Push Pattern with Amazon S3

This architecture diagram shows data ingestion and a push pattern for data available in SaaS applications.



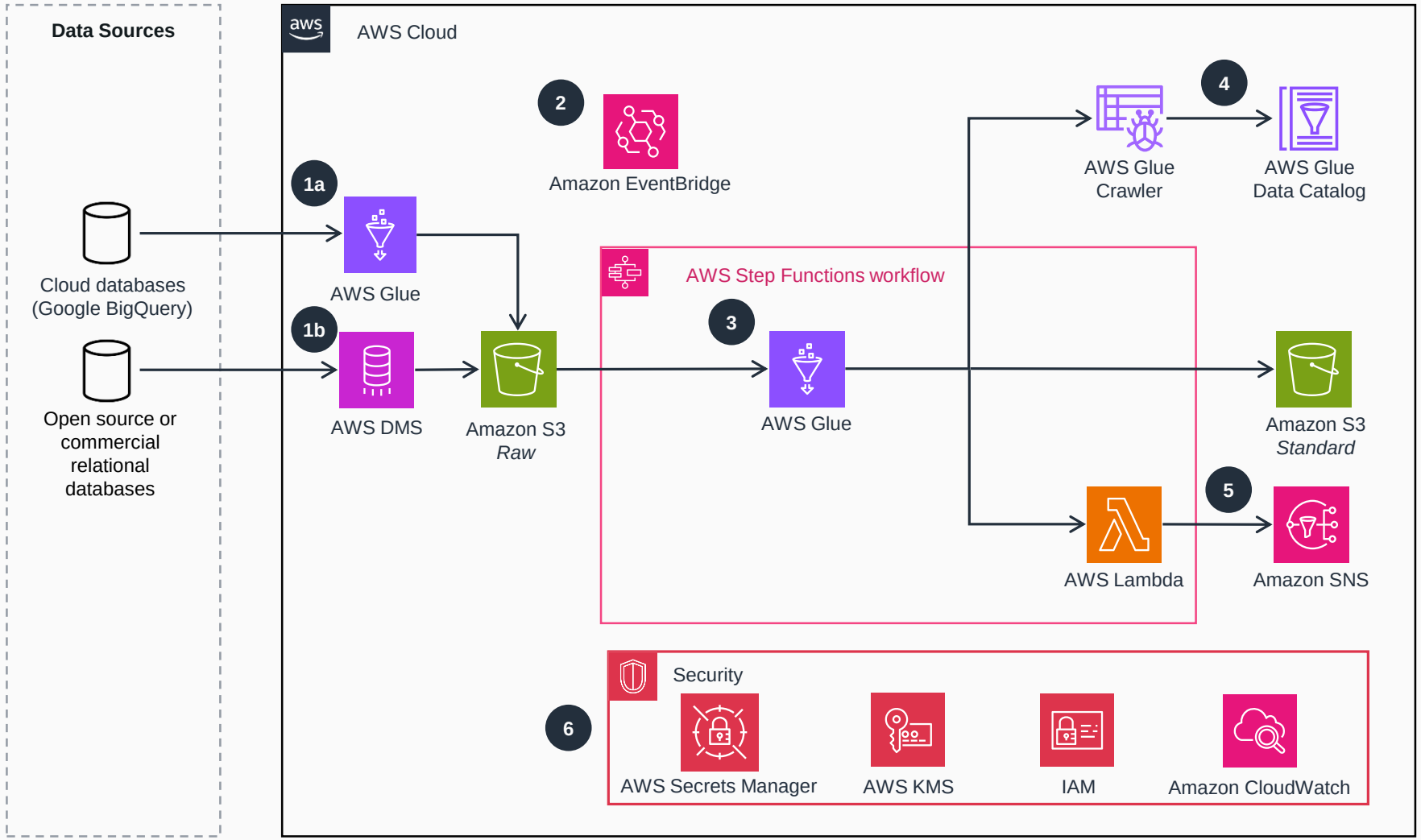
1. External data sources push raw data files (such as CSV) into a daily partitioned Landing Zone **S3** bucket. Refer to external documentation to set up push job inputs like **S3** bucket location, access key, and schedule frequency.
2. Create a rule in **EventBridge** to schedule a **Step Functions** standard workflow for data processing at required frequency.
3. In the workflow, use a **Lambda** function to do file-level processing, such as Pretty Good Privacy (PGP) decryption. Place the decrypted file in a different **S3** bucket prefix.
4. Use **AWS Glue** jobs to process the decrypted data files in the Landing Zone **S3** bucket, and write data in a separate Processed Zone **S3** bucket. Write the object in read optimized Apache Parquet format. Apply attribute level transformation like SHA256 hashing to secure sensitive data. Apply partitioning scheme as needed to optimize reads.
5. The **AWS Glue crawler** executes from the workflow to catalog the read optimized data in the **Data Catalog**.
6. Use a **Lambda** function to do post processing activities, such as moving the source data files to an "archive" prefix location as part of clean-up.
7. Use **Amazon SNS** to publish a workflow complete event and notify operators and users using email. Use HTTP or Topic options to integrate with other observability tools.
8. Use the following AWS services for security and access: **IAM** enables least privilege access to specific resources and operations. **AWS KMS** provides encryption for data at rest and data in transit. **Secrets Manager** provides hashing keys for PII data. **CloudWatch** monitors logs and metrics across all services used in this Guidance.

Reviewed for technical accuracy February 19, 2024

AWS Reference Architecture

Connecting RDBMS Sources to AWS – Batch Pull and Change Data Capture Pattern

This architecture diagram shows how to build a connector for relational database management systems (RDBMS) to AWS.



1a Use **AWS Glue** and pre-built or marketplace connectors to extract data needed for advertising and marketing analytical use case from the relational database management system (RDBMS) in batch mode. **AWS Glue** will retrieve the data from data stores and load it to an **S3** bucket**. Amazon S3** is configured as a target for storing remote database files in parquet format.

1b Use **AWS Database Migration Service (AWS DMS)** to replicate data stored in compatible relational databases (on-premises or on a cloud) to AWS.

2 A rule in **EventBridge** schedules a **Step Functions** standard workflow for post upload processing at required frequency.

3 **AWS Glue** jobs and workflows do the row-level processing of the decrypted data files and write data in a separate **S3** bucket. Write the object in read optimized Apache Parquet format. Apply attribute level transformation like SHA256 hashing to secure sensitive data. Apply custom partitioning scheme as needed to optimize reads.

4 The **AWS Glue crawler** executes from the workflow to catalog the read optimized data in the **Data Catalog**.

5 Publish a notification to **Amazon SNS** and notify operators of success or failure of the workflow. Use the HTTP or Topic option to integrate with other observability tools.

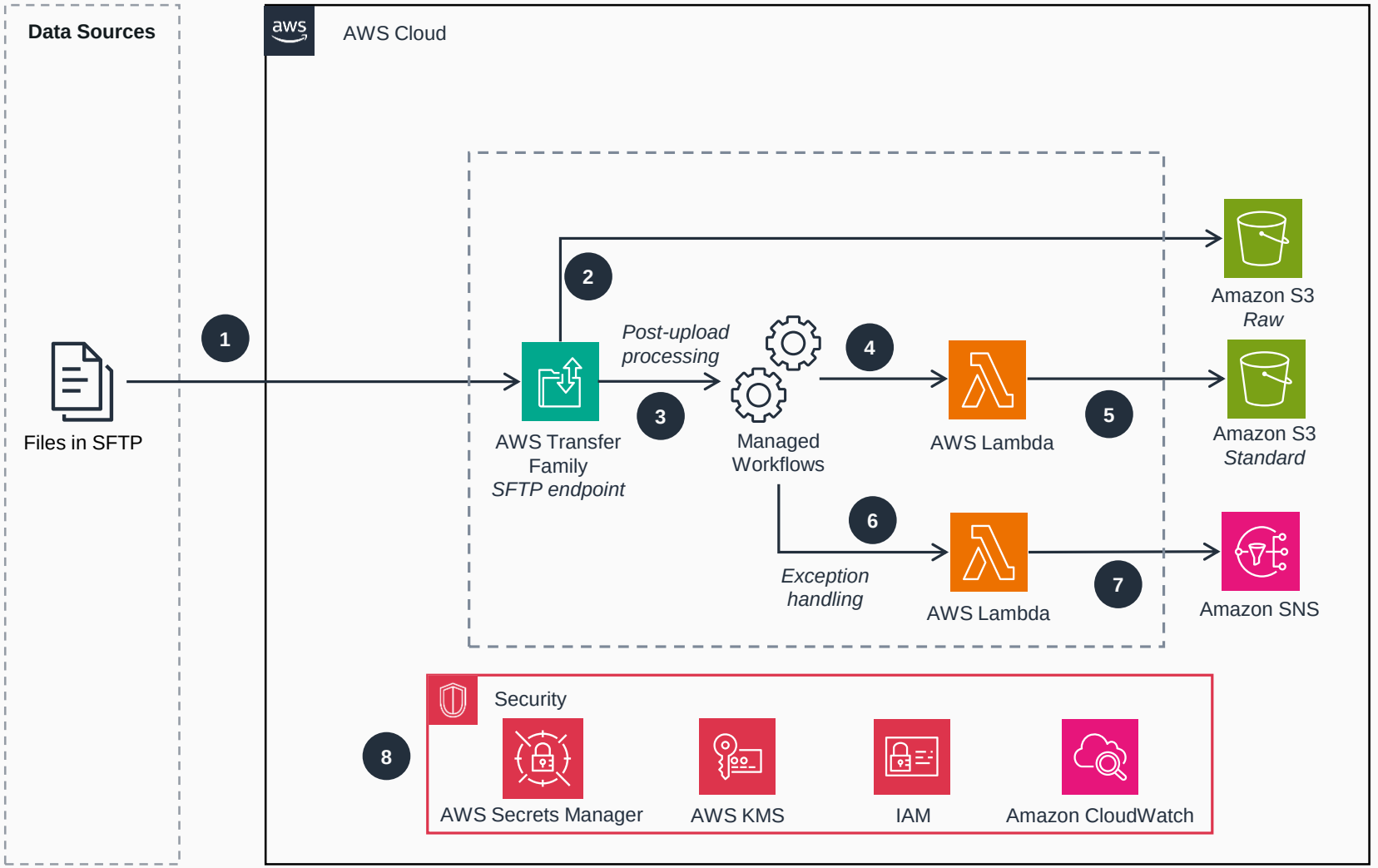
6 Use the following AWS services for security and access: **IAM** enables least privilege access to specific resources and operations. **AWS KMS** provides encryption for data at rest and data in transit. **Secrets Manager** provides hashing keys for PII data. **CloudWatch** monitors logs and metrics across all services used in this Guidance.

Reviewed for technical accuracy February 19, 2024

AWS Reference Architecture

Connecting SFTP Data Sources to AWS – Managed File Transfer Pattern

This architecture diagram shows how to build a connector for file systems to AWS.



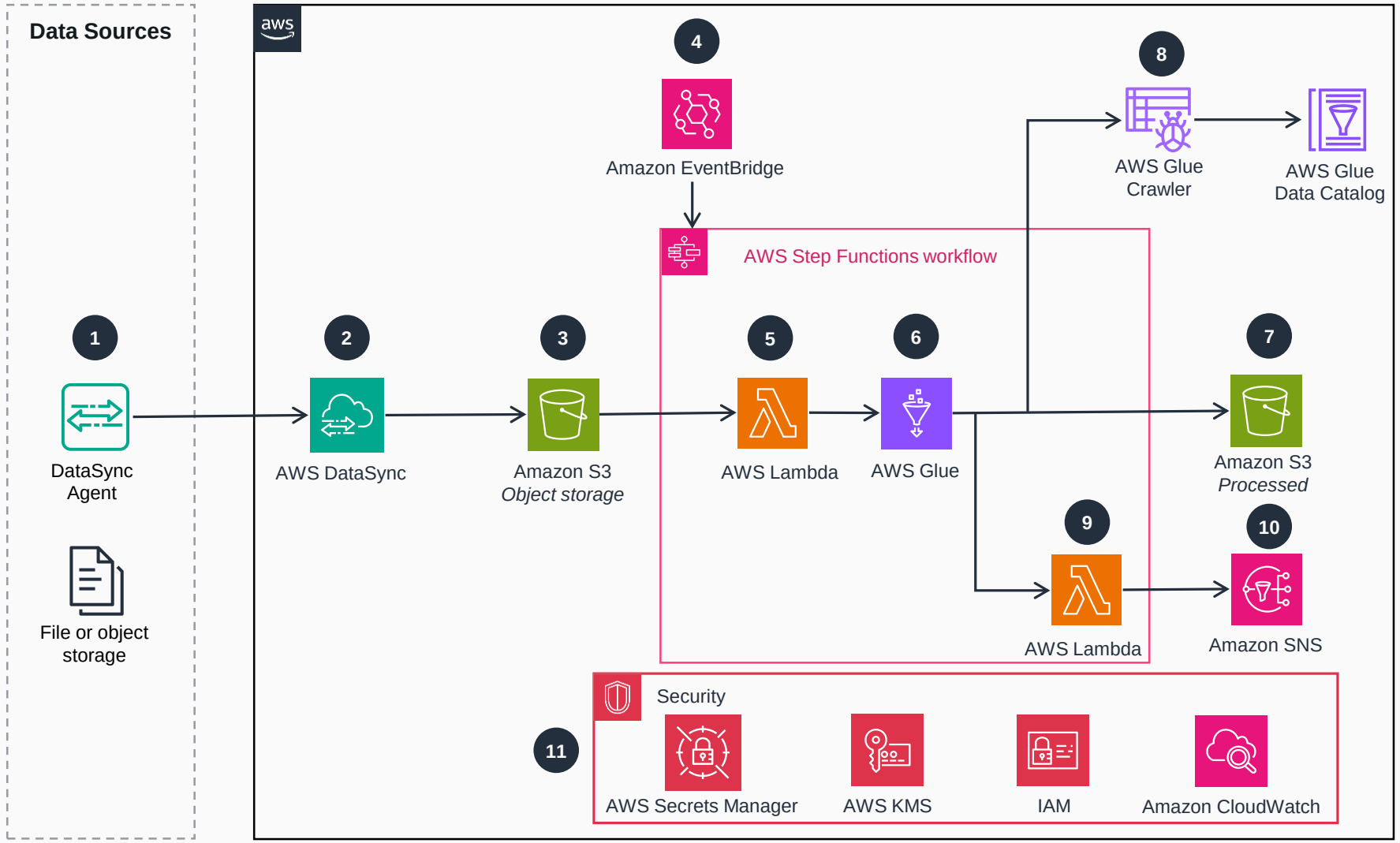
1. **AWS Transfer Family** securely migrates data stored in file systems (on-premises or on a cloud) that is needed for advertising and marketing analytical use cases.
2. Raw files from remote servers are uploaded as-is into a raw **S3** bucket.
3. **Transfer Family managed workflows** complete post-upload processing of files, including decrypting, error checking, and formatting changes.
4. **Lambda** completes custom post processing of data before sending it to storage.
5. Processed data that is ready for analysis by applications and other data consumers is stored in a standard **S3** bucket.
6. **Transfer Family** managed workflows will invoke a **Lambda** function when post upload processing steps fail for a file
7. **Amazon SNS** publishes an exception event to notify users through email or other observability tools.
8. Use the following AWS services for security and access: **IAM** enables least privilege access to specific resources and operations. **AWS KMS** provides encryption for data at rest and data in transit. **Secrets Manager** provides hashing keys for PII data. **CloudWatch** monitors logs and metrics across all services used in this Guidance.

Reviewed for technical accuracy February 19, 2024

AWS Reference Architecture

Connecting File and Cloud Object Storage to AWS – File Replication Pattern

This architecture diagram shows how to build a connector for cloud-based object storage services to AWS.



1. Install and configure **AWS DataSync Agent** on a virtual machine in the public cloud where the source Object Storage is hosted.
2. The **DataSync Agent** and **DataSync** allow discovery and scheduling of data transfer for both the initial sync and continuous, ongoing sync.
3. Configure **DataSync** to store the replicated data in a Landing Zone **S3** bucket.
4. Create a rule in **EventBridge** to schedule a **Step Functions** standard workflow for data processing at required frequency.
5. In the workflow, use a **Lambda** function to do any necessary file- or object-level decryption, and invoke an **AWS Glue** task to normalize the data.
6. Use **AWS Glue** jobs and workflows to do data processing of the decrypted data files, and write data in a separate **S3** bucket.
7. Write the object in read optimized Apache Parquet format. Apply data. Apply custom partitioning scheme as needed to attribute level transformation like SHA256 hashing to secure sensitive optimize reads.
8. Create an **AWS Glue crawler**, and add it to the workflow to catalog the read optimized data in **Data Catalog**.
9. Use another **Lambda** function to do post-processing activities, such as moving the source data files to an "archive" prefix location as part of clean-up and to save on storage costs.
10. Use **Amazon SNS** to publish a workflow complete event and notify operators and users using email. Use HTTP or Topic option to integrate with other observability tools.
11. Use the following AWS services for security and access: **IAM** enables least privilege access to specific resources and operations. **AWS KMS** provides encryption for data at rest and data in transit. **Secrets Manager** provides hashing keys for PII data. **CloudWatch** monitors logs and metrics across all services used in this Guidance.

Reviewed for technical accuracy February 19, 2024

AWS Reference Architecture