AWS Whitepaper

Patterns for Ingesting SaaS Data into AWS Data Lakes

Patterns for Ingesting SaaS Data into AWS Data Lakes: AWS Whitepaper

Table of Contents

Patterns for Ingesting SaaS Data into AWS Data Lakes

Publication date: **March 4, 2022** (*Document Revisions*)

Abstract

Today, many organizations generate and store data in Software as a service (SaaS)-based applications. They want to ingest this data in a central repository (often a data lake) so that they can analyze it though a single pane of glass and derive insights from it. However, because different SaaS applications provide different mechanisms of extracting the data, a single approach does not always work.

This paper outlines different patterns using Amazon Web Services (AWS) services to ingest SaaS data into a data lake on AWS.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems on AWS. Using the Framework allows you to learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud.

In the SaaS Lens, we enable customers to review and improve their cloud-based architectures and better understand the business impact of their design decisions. We address general design principles as well as specific best practices and guidance in five conceptual areas that we define as the pillars of the Well-Architected Framework.

Introduction

With the growing ecosystem of SaaS-based applications, many organizations face the challenge of collecting all this data into a centralized location for analytics. Often this centralized location is a data lake, and in AWS Cloud, Amazon Simple Storage Service (Amazon S3) becomes the centralized storage for this data lake. However, an ingestion mechanism must be in place get all the SaaS data into Amazon S3. This paper explores different options to use in AWS, along with usage patterns and considerations to make when selecting a particular ingestion mechanism.

Data ingestion mechanisms

In this section, we'll review the following data ingestion mechanisms: Purpose-built integration service, ETL using custom connectors with Apache Spark, data federation using SQL engine, streaming data, and import/export using files.

Purpose-built integration service

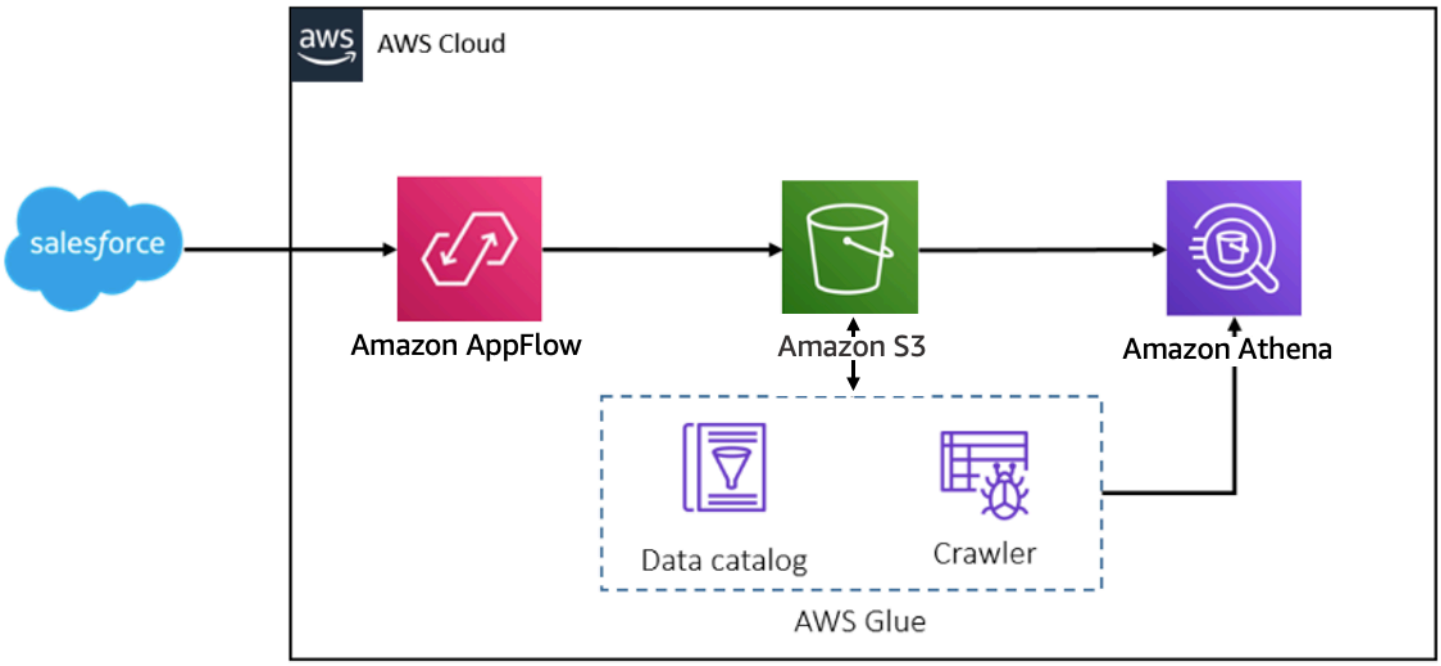
Amazon AppFlow: Introduction

Amazon AppFlow is a fully-managed integration service that enables you to securely transfer data between SaaS applications (such as Salesforce, Marketo, Slack, and ServiceNow) and AWS services (such as Amazon S3 and Amazon Redshift). With Amazon AppFlow, you can run data flows at nearly any scale and frequency (on a schedule, in response to a business event in real time, or on demand). You can configure data transformations such as data masking and concatenation of fields, as well as validate and filter data (omitting records that don't fit a criteria) to generate rich, ready-to-use data as part of the flow itself, without additional steps.

Amazon AppFlow automatically encrypts data in motion, and optionally allows you to restrict data from flowing over the public internet for SaaS applications that are integrated with AWS PrivateLink, reducing exposure to security threats. For a complete list of all the SaaS applications that can be integrated with Amazon AppFlow, refer to Amazon AppFlow integrations.

Architecture overview

The following diagram depicts the architecture of the solution where data from Salesforce is ingested into Amazon S3 using Amazon AppFlow. Once the data is ingested in Amazon S3, you can use an AWS Glue crawler to populate the AWS Glue Data Catalog with tables and start consuming this data using SQL in Amazon Athena.



Amazon AppFlow-based data ingestion pattern

Usage patterns

Because Amazon AppFlow can connect to many SaaS applications and is a low-code/no-code approach, this makes it very appealing for those who would want a quick and easy mechanism to ingest data from these SaaS applications.

Some use cases are as follows:

- Create a copy of a Salesforce object (for example, opportunity, case, campaign) in Amazon S3.
- Send case tickets from Zendesk to Amazon S3.
- Hydrate an Amazon S3 data lake with transactional data from SAP S/4HANA enterprise resource planning (ERP).
- Send logs, metrics, and dashboards from Datadog to Amazon S3, to create monthly reports or perform other analyses automatically, instead of doing this manually.
- Send Marketo data, like new leads or email responses, in Amazon S3.

Considerations

If a SaaS application you want to get data from is not supported out of the box, you can now build your own connector using the Amazon AppFlow custom connector SDK. AWS has released the

python custom connector SDK as well as java custom connector SDK for AppFlow. The AppFlow Custom Connector SDK enables customers and third-party developers to build custom source and/or destination connectors for the AppFlow service. With the SDK, you can connect to private APIs, on-premise proprietary systems, and other cloud services by adding to AppFlow's library of connectors.

If any of the following scenarios apply, other ingestion patterns discussed in this paper may be a better fit for your type of ingestion:

- A supported application is heavily customized.
- Your use case exceeds any of the application-specific limitations.

For every SaaS application that Amazon AppFlow supports, there are a set of limitations included. For example, if you are transferring more than one million Salesforce records, you cannot choose any Salesforce compound field. Before using Amazon AppFlow, look for the limitations for the application that you are planning to connect, evaluate your use case against those limitations, and see if the service is still a good fit for what you are trying to do.

SaaS applications are sometimes heavily customized, so it's always good to make sure the edge cases can be solved with Amazon AppFlow. You can find the list of known limitations and considerations in the notes section of the Amazon AppFlow documentation. For example, the known limitations for Salesforce as a source are listed here.

Also, consider the Amazon AppFlow service quotas to ensure your use case fits well within those limitations.

Extract, transform and load (ETL) using custom connectors with Apache Spark

AWS Glue: Introduction

AWS Glue is a serverless data integration service that makes it easy to discover, prepare, and combine data for analytics, machine learning, and application development. AWS Glue provides all the capabilities needed for data integration so that you can start analyzing your data and putting it to use in minutes instead of months.

AWS Glue custom connectors make it easy to discover and integrate with a variety of additional data sources, such as SaaS applications or your custom data sources. With just a few clicks, you can

search and select connectors from the AWS Marketplace and begin your data preparation workflow in minutes. AWS is also releasing a new framework to develop, validate, and deploy your own custom connectors (bring your own connectors (BYOC)).

Architecture overview

The following diagram depicts the architecture of the solution where data from any of the SaaS applications can be ingested into Amazon S3 using AWS Glue. Once the data is ingested in Amazon S3, you can catalog it using AWS AWS Glue Data Catalog, and start consuming this data using SQL in Amazon Athena.

AWS Glue-based data ingestion pattern

Usage patterns

Because AWS Glue ETL provides the data engineers with flexibility, scalability, and the open-source distributed computing power of Apache Spark framework, this option is an excellent choice for ingesting SaaS data into the AWS Cloud ecosystem (including Amazon S3 for data lake operations).

Some use cases are as follows:

- Enrich internal system data with data from SaaS applications during the ETL process.
- Look up transformations with data from SaaS applications during ETL process.

The flexibility can be applicable in multiple places during the ETL process using AWS Glue:

- You can either use the existing connectors that can be found in the AWS Marketplace, or if you don't see a pre-built connector for the SaaS application you are trying to connect to, then you can also build your own customer connector and use that in AWS Glue. The process to create your own connector is documented here: Developing custom connectors.
- A connector is just an initial piece of the puzzle which allows you to connect to the source application. But often, the data that you want from the source may need to be transformed, normalized, or aligned in a specific format before it can be stored in Amazon S3. This is where AWS Glue shines: It provides the freedom to data engineers to customize their ETL process to cater to the complexities as defined by their use cases. The complexities may arise due to heavy customizations to the SaaS application, they may arise due to limitations of other low-code/no-code tools, or they may arise due to complex business/technical requirements around how the data should be persisted in the target storage.

AWS Glue Studio is a new graphical interface that makes it easy to create, run, and monitor ETL jobs in AWS Glue. You can visually compose data transformation workflows and seamlessly run them on AWS Glue's Apache Spark-based serverless ETL engine. Using AWS Glue Studio and AWS Marketplace connectors, you can create an ETL pipeline with ease. For example, refer to Migrating data from Google BigQuery to Amazon S3 using AWS Glue custom connectors.

Considerations

Using custom connectors in AWS Glue Studio has some limitations. However, data engineers have the option to author jobs using the AWS Glue console and write their code themselves. They can use the custom connectors or use other open-source/paid connector codes to access any SaaS application. For example, refer to Extract Salesforce.com data using AWS Glue and analyzing with Amazon Athena.

Also, keep in mind when using the AWS Glue ETL pattern for ingesting SaaS data, the source SaaS applications may have different limitations built in for bulk data extraction as well as API limits. For

example, Salesforce has these limits for bulk data extraction: Bulk API and Bulk API 2.0 Limits and Allocations.

Data federation using SQL engine

Amazon Athena: Introduction

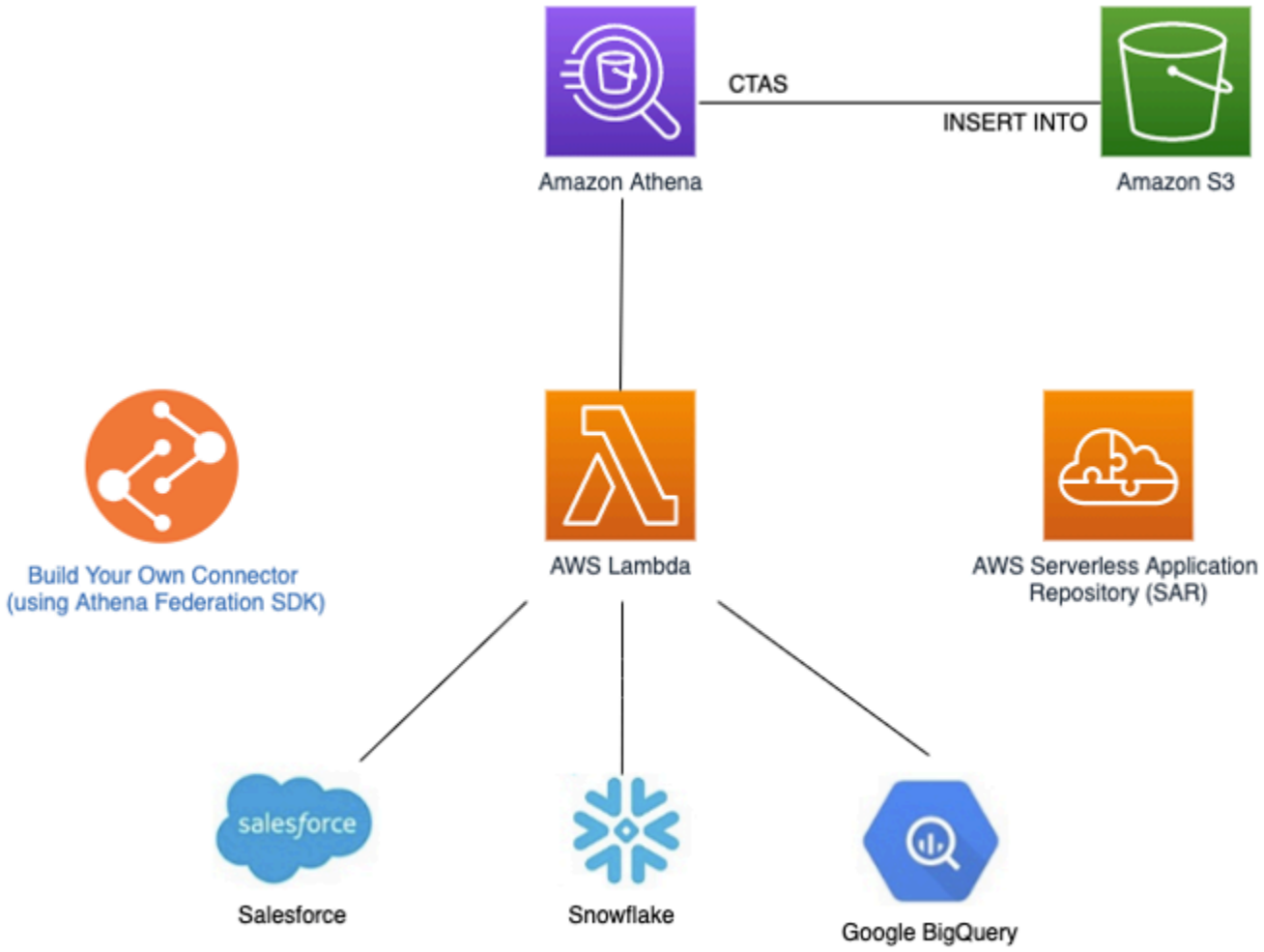
Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run.

Amazon Athena Federated Query is a new Athena feature that enables data analysts, engineers, and data scientists to run SQL queries across data stored in relational, non-relational, object, and custom data sources. With Amazon Athena Federated Query, customers can submit a single SQL query and analyze data from multiple sources running on-premises or hosted on the cloud. Athena runs federated queries using Data Source Connectors that run on AWS Lambda.

Customers can use these connectors to run federated SQL queries in Athena across multiple data sources, including SaaS applications. Additionally, using Query Federation SDK, customers can build connectors to any proprietary data source, and enable Athena to run SQL queries against the data source. Because connectors run on Lambda, customers continue to benefit from Athena's serverless architecture and do not have to manage infrastructure or scale for peak demands.

Architecture overview

Athena Federated query connector allows Athena to connect to SaaS applications like Salesforce, Snowflake, and Google BigQuery. Once a connection is established, you can write SQL queries to retrieve data stored in these SaaS applications. To store data in Amazon S3 data lake, you can use Athena statements Create Table as Select (CTAS) and INSERT INTO for ETL, which then store the data in Amazon S3 and create a table in the AWS AWS Glue Data Catalog.



Amazon Athena-based data ingestion pattern

Usage patterns

This pattern is ideal for anyone who can write ANSI SQL queries. Also, the Create Table as Select (CTAS) statement of Athena provides the flexibility to create a table in AWS AWS Glue Data Catalog with the selected fields from the query, along with the file type for the data to be stored in Amazon S3. You can do an ETL transformation with ease, and the final datasets can be made ready for end user consumption. You can also use AWS Step Functions to orchestrate the whole ingestion process. For details, refer to Build and orchestrate ETL pipelines using Amazon Athena and AWS Step Functions.

Some use cases are as follows:

- Analysts need to create a real-time, data-driven narrative, and identify specific data points.

- Administrators need to deprecate old ingestion pipelines, and move to a serverless ingestion pipeline using just a few SQL queries.
- Data engineers do not need to learn different data access paradigms; they can use SQL to source data from any particular data source.
- Data scientists needs to use data from any SaaS data source to train their machine learning (ML) models, and that helps them to improve the accuracy of their ML models as well.

Considerations

Some third-party Athena connectors found in the AWS Serverless Application Repository are paid connectors, so due diligence should be done on the pricing aspect of such SaaS connectors. Also, SaaS applications may have their own bulk export limits and/or data export charges that need to be accounted for.

Streaming data

Amazon Managed Streaming for Apache Kafka and Amazon Kinesis: Introduction

Amazon Managed Streaming for Apache Kafka (Amazon MSK) makes it easy to ingest and process streaming data in real time with fully-managed Apache Kafka.

Amazon Data Firehose is an ETL service that reliably captures, transforms, and delivers streaming data to data lakes, data stores, and analytics services.

Architecture overview

Often, use cases require the source data to be to immediately made available in the data lake for analytics. Getting the data from the SaaS application in near-real-time lays the foundation for an event-driven architecture. Start with how Apache Kafka helps with this architecture pattern: Apache Kafka has a tool called Kafka Connect, which allows for scalable and reliable streaming data between Apache Kafka and other systems. Many SaaS applications allow for events data to be published to Kafka topics using Kafka Connect.

Amazon MSK eliminates the operational overhead, including the provisioning, configuration, and maintenance of highly-available Apache Kafka and Kafka Connect clusters. With Amazon MSK Connect, a feature of Amazon MSK, you can run fully-managed Apache Kafka Connect workloads on AWS. This feature makes it easy to deploy, monitor, and automatically scale connectors that

move data between Apache Kafka clusters and external systems. Amazon MSK Connect is fully-compatible with Kafka Connect, enabling you to lift and shift your Kafka Connect applications with zero code changes. With Amazon MSK Connect, you only pay for connectors you are running, without the need for cluster infrastructure.

Once the data is streamed in Kafka, you can either directly send it to Amazon S3 or you can use the Kinesis-Kafka Connector to extend the pipeline before the data is pushed into Amazon S3. The advantage of using Firehose in the streaming data pipeline is that you can now transform/normalize the events data, buffer it for a specific interval or a specific file size, and finally convert this data into Parquet file format before it lands in Amazon S3. You can also partition the data in Amazon S3 prefixes based on rules using Firehose.

Streaming based-data ingestion pattern

Usage patterns

This pattern is handy when you want to stream data changes from SaaS applications into your Amazon S3 data lake in near-real-time.

Some use cases are as follows:

- Real-time analytics dashboards.
- External, API-driven applications.

Considerations

The above architecture pattern adds all records in Amazon S3 as new data. To handle updates and deletes in the data lake, you may have to use AWS Glue, streaming with Apache Hudi connector, to correctly identify and update the existing records in Amazon S3.

Import/Export using files

Many SaaS applications allow data to be exported into files. You can use any of the file transfer mechanisms listed below to move such files into Amazon S3 for further processing:

AWS DataSync is a secure, online service that automates and accelerates the process of moving data between on premises and AWS storage services. DataSync can copy data between Network File System (NFS) shares, Server Message Block (SMB) shares, Hadoop Distributed File Systems (HDFS), self-managed object storage, AWS Snowcone, Amazon S3 buckets, Amazon Elastic File System (Amazon EFS) file systems, Amazon FSx for Windows File Server file systems, and Amazon FSx for Lustre file systems.

AWS Transfer Family securely scales your recurring business-to-business file transfers to Amazon S3 and Amazon EFS using SFTP, FTPS, and FTP protocols.

You can also write your own scripts using AWS Command Line Interface (CLI) or AWS SDK to transfer the exported files over to Amazon S3 at a regular interval.

Conclusion

In the ever-growing world of SaaS-based applications, the data produced and stored by these applications will continue to grow exponentially in the future. This paper looked at some of the mechanisms with which you can ingest this data into Amazon S3, and build a central data lake. Based on your use case, conduct a proof of concept with the best fitting mechanism listed in this paper, to make sure all the edge cases can be met and you get a cost effective, agile, and scalable system.

Contributors

Contributors to this document include:

- Behram Irani, Senior Solutions Architect, Amazon Web Services

Further reading

For additional information, refer to:

- Hydrate your data lake with SaaS application data using Amazon AppFlow
- Query Snowflake using Athena Federated Query and join with data in your Amazon S3 data lake
- Introducing Amazon MSK Connect – Stream Data to and from Your Apache Kafka Clusters Using Managed Connectors
- Developing, testing, and deploying custom connectors for your data stores with AWS Glue

Document Revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Whitepaper published.	March 4, 2022

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.