

Extracting data from SAP in a plug-and-play way with Amazon AppFlow

4 December 2024 - 2 min. read

[Data and Analytics](#)

[Data Lake](#)

In a world where data drives business innovation, the ability to collect, organize, and analyze it efficiently is crucial. A data lake is a centralized repository that ingests and stores large volumes of data in its original form. Thanks to its scalable architecture, a data lake can accommodate all types of data from any source – structured (relational database tables, csv files), semi-structured (XML files, webpages), and unstructured (images, audio files, tweets) – all without sacrificing fidelity.

AWS offers a comprehensive ecosystem of services that enable you to create and manage data lakes, making them an ideal foundation for a wide range of analytic needs. Additionally, through integrations with third-party services, it's possible to further enhance a data lake, gaining a holistic and in-depth view of business data.

In this article, we will explore a specific integration between Amazon AppFlow and SAP to prepare data for a data analysis platform.

Let's start defining the main actors in our scenario.

What is Amazon AppFlow?

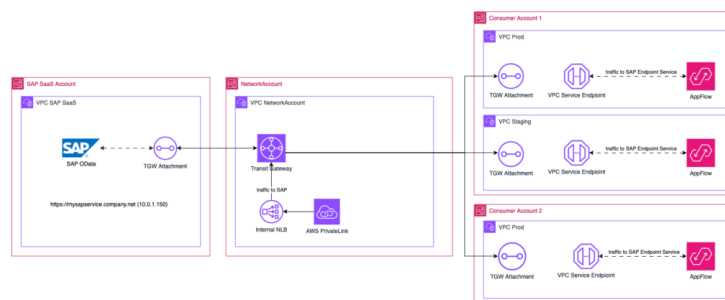
Imagine being able to transfer data between your SaaS applications and AWS quickly, securely, and without implementing complex automation. That's the magic that Amazon AppFlow offers: a service that simplifies integrations and allows you to automate data flows with just a few clicks, taking care of all the important aspects of data flow integrations. It relieves you of the duties of managing complex infrastructure and writing code to integrate different data sources.

A crucial aspect of data flow integration is the privacy of the connection channel used. Fortunately, AWS helps address this concern with its PrivateLink service. Amazon AppFlow natively integrates with **AWS PrivateLink**, enabling connections to data sources through a private connection without exposing data to the public internet. This ensures enhanced data security, especially when dealing with sensitive information.

Prerequisites and AWS Organization Layout

Let's analyze our AWS Organization's setup focusing on the main aspects:

- **A management account** responsible for centralized networking, acting as a central hub for all other accounts within the organization. We have set up a Transit Gateway in this account and shared it with the organization. This enables us to connect our main account to the account owned by SAP.
- **Multiple child accounts** within our organization that will be the consumers of the data residing in SAP Cloud.
- **An external AWS account owned by SAP.** This is where our data is located.



Please note that the SAP-owned account, as well as our accounts are all connected through Transit Gateway VPC Attachments. This scalable network setup ensures that all our current and future AWS accounts will be able to communicate between each other. Another aspect that you should consider is the proper configuration of the routing: following the best practice of least privilege guarantees that accounts have access only to the data and services they are explicitly permitted to use.

Setup Endpoint Service

Once the SAP services are connected to the main account we need to find an effortless way to make the data available to our consumers. For this purpose, we can use an Endpoint Service powered by AWS PrivateLink

PrivateLink enables us to share a service hosted in AWS with other consumer accounts, it requires a Network Load Balancer (supports also Gateway Load Balancers), which receives requests from consumers and routes them to your service.

In order to configure a private Amazon AppFlow connection, we must first create and verify an Endpoint Service that will be the entry point to the SAP services.

To create the Endpoint Service, you need to consider various things:

- **A DNS Name.** You need to define the private DNS Name that will be used for the Endpoint Service. Note that the DNS name should be located in a public Hosted Zone for you to be able to validate the ownership.
- **ACM Certificate for the domain.** Consider requesting the certificate for a wildcard domain to prevent potential issues regarding subdomain mismatches.
- **Network Load Balancer, Target Group, and SAP endpoint details.** You need to create an internal Network Load Balancer to associate with the Endpoint Service. You also need some basic information about the targets to register them to the Target Group (e.g., port, protocol, IP addresses). In our specific case, we are dealing with a TLS service on port 44300.
- **Validate the Private DNS Name.** You will be asked to confirm the ownership of the selected domain via a TXT record.

Although these steps are more focused on SAP services, you can apply the same principles to any other external services that you want to integrate with your consumer applications in AWS.

Wait a few minutes for provisioning and validation to take effect and you should be ready to go!

Setup Amazon AppFlow Connection

Now that we have an Endpoint Service, we can move to the child AWS accounts and create a new Flow that has a SAP OData source.

From the Amazon AppFlow console select SAP OData as Connector type and insert the required parameters for the connection:

- **SAP OData Connector type**
- **Application Host URL.** This is the custom DNS name that you defined during the setup of the Endpoint (e.g., <https://mysapservice.example.com>)
- **Application service path, port number and client number.** May vary depending on your services (e.g., path: `/sap/opu/odata/iwfnd/catalogservice;v=2`, port number: 44300, client number: 100)

- **PrivateLink.** Set to “Enabled” and insert the AWS PrivateLink service name. You can find it in the VPC Endpoint Service dashboard. (e.g., **com.amazonaws.vpce.eu-west-1.vpce-svc-12341234333344455**)
- **Authentication mode:** Basic Auth. Note that OAuth2 authentication is not feasible for our use-case as it requires user interaction via browser and cannot be done via PrivateLink.
- **Username and Password.** Amazon AppFlow will automatically create a Secrets Manager Secret for your service credentials so you can change them later if needed.

Once all fields are filled save and test the connection to SAP.

Next Steps

With the help of Amazon AppFlow and a VPC Endpoint Service, we have achieved a private connection to our datastore in SAP and can now create a new data flow.

For example you can set up a Flow with an S3 Bucket as a destination, you can configure it to run on demand or on a recurring schedule.

You can choose json or csv as output formats, or even better, you can choose parquet format. Using parquet allows you to have highly compressed and optimized data storage, which in turn allows you to optimize storage costs and data querying costs with AWS Athena.

Conclusion

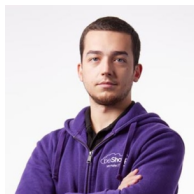
In this article we have covered only the SAP OData Connector, but it is just one of the many integrations that you can set up with Amazon AppFlow. We just scratched the surface of what can be done and the vast possibilities that a data lake gives you. From this starting point you can expand your data lake and ingestion system to accommodate many more integrations with external heterogeneous data sources. This enables you to start analyzing, transforming, and gaining valuable insights from your data from a centralized point of view.

Feel free to explore all the customization options that Amazon AppFlow gives you to find the configuration that fits best to your use case!

About Proud2beCloud

Proud2beCloud è il blog di **beSharp**, APN Premier Consulting Partner italiano esperto nella progettazione, implementazione e gestione di infrastrutture Cloud complesse e servizi AWS avanzati. Prima di essere scrittori, siamo Solutions Architect che, dal 2007, lavorano quotidianamente con i servizi AWS. Siamo innovatori alla costante ricerca della soluzione

più all'avanguardia per noi e per i nostri clienti. Su Proud2beCloud condividiamo regolarmente i nostri migliori spunti con chi come noi, per lavoro o per passione, lavora con il Cloud di AWS. Partecipa alla discussione!



Mehmed Dourmouch

DevOps Engineer. Very Dev, not so Ops. I like to break things and see what happens, I also automate everything. I often participate in cybersecurity CTFs and in my free time I produce cacophony with my guitar.

Copyright © 2011-2024 by beSharp spa - P.IVA IT02415160189