

Implementation Guide:

Automate Snowflake integration with Amazon S3 using AWS Service Catalog in AWS Marketplace



Table of Contents

Foreword	3
Solution overview and features	3
Architecture diagram	3
Pre-requisites	5
Deployment and Configuration Steps	5
Step 1: Install	5
Step 2: Test and Run	6
Solution Estimated Pricing	7
FAQs	7
Additional resources	7
Partner contact information	7

Foreword

[Snowflake](#) is the data warehouse built for the cloud that provides enterprises with a cost-effective combination of the power of data warehousing, the flexibility of big data platforms, and the elasticity of the cloud. Snowflake is a member of the AWS Partner Network (APN) and [Snowflake available from the AWS Marketplace](#) helps you bring together diverse data into one system for data warehousing, query processing, and analytics.

Solution overview and features

[Snowflake storage integrations](#) are Snowflake objects that allow Snowflake to read and write data to Amazon S3. Snowflake storage integrations leverage [AWS IAM](#) to access S3. The S3 bucket is referenced by the Snowflake integration from an external (i.e. S3) [Snowflake stage object](#).

The solution implements a Service Catalog product that automates Snowflake access to S3. The Service Catalog product provisions the Snowflake integration object, attaches an IAM role to it and creates a Snowflake stage object for it that references S3. The Service Catalog product uses [AWS Secrets Manager](#) to store and retrieve Snowflake connection information.

Each time, the Service Catalog product is launched, it creates a new Snowflake external stage object to access an S3 object/prefix based on 2 input parameters – Snowflake connection information and the S3 bucket (with prefix). The solution is fully automated using AWS CloudFormation.

Architecture diagram

1. The Service Catalog administrator installs the solution by launching the [aws-snowflakeintobj-servicecatalog](#) CloudFormation template. This template provisions a [Service Catalog Portfolio](#) with a Service Catalog Product.
2. The Service Catalog end user (or administrator) launches the Snowflake Service Catalog Product. This product takes a) Snowflake Connection information and b) S3 bucketname and prefix as input parameters and uses the [aws-snowflake-integrationobject.yml](#) CloudFormation template to create a Snowflake external stage object that enables access to S3.
 - i. The Snowflake Service Catalog Product can be launched as many times as needed. Each time it creates a new Snowflake external stage object to access a new S3 object/prefix based on the input parameters (a and b) supplied above.
3. The Service Catalog Product in Step 2 above:

- i. Provisions Secrets Manager to store and retrieve Snowflake connection information
- ii. Provisions a Lambda function that uses the Snowflake python connector:
 - a. Creates a Snowflake integration object and obtains the Snowflake generated `AWS_IAM_USER_ARN` and `AWS_EXTERNAL_ID` from the Snowflake integration
 - b. Provisions an AWS IAM role that uses the Snowflake generated `AWS_IAM_USER_ARN` and `AWS_EXTERNAL_ID` in the [trust policy](#) of the IAM role
 - c. Creates a Snowflake stage object that leverages the snowflake integration

The following architecture diagram illustrates the components of the Service Catalog, Snowflake and S3 integration

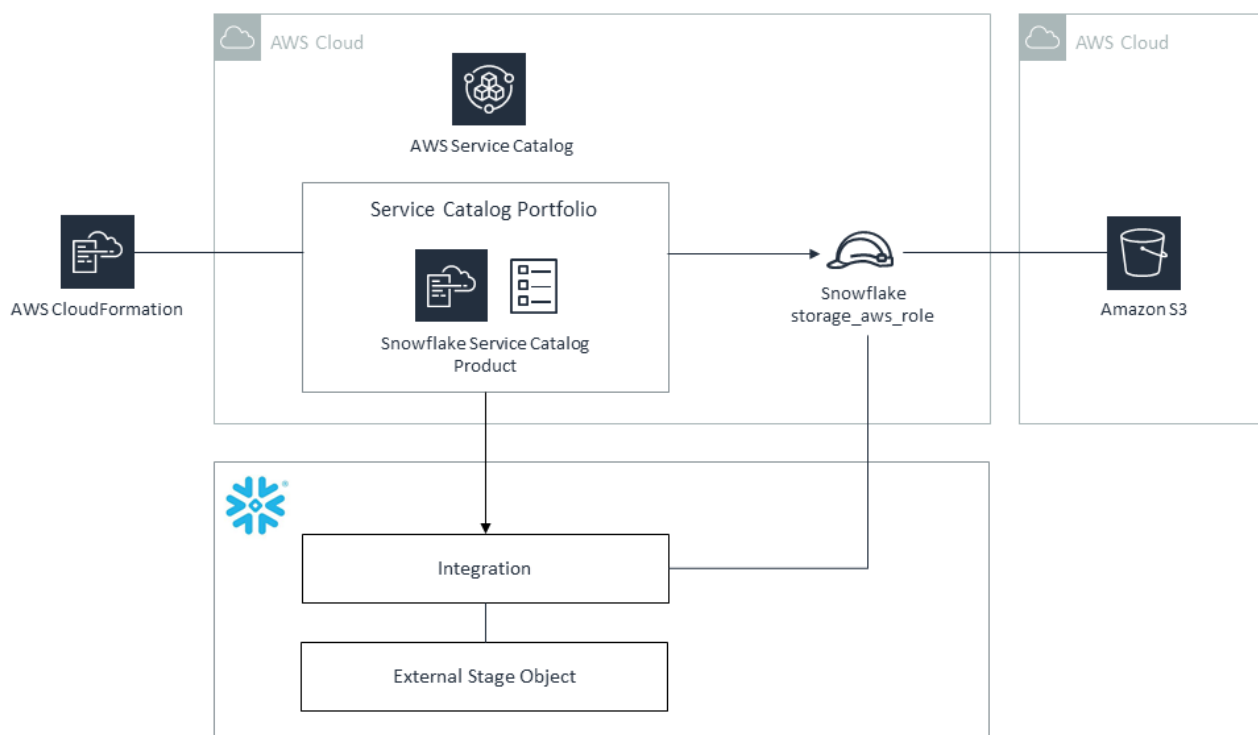


Figure 1 Snowflake Storage Integration Architecture Diagram

Pre-requisites

You must complete the following prerequisites before implementing the Snowflake and S3 automation using Service Catalog:

1. Subscribe to Snowflake via [AWS Marketplace](#). In the upper right, choose **Continue to Subscribe**. After successfully subscribing to Snowflake, choose **Accept Terms**. To begin using the software, you will be redirected to the Snowflake website. Select the **Set Up Your Account** button to sign up with Snowflake and complete your registration.
2. Once you are set up with your Snowflake account and if you are new to Snowflake, then complete this excellent [Snowflake in 20 minutes](#) tutorial. At the end of this tutorial you will learn how to create required Snowflake objects (warehouses, databases and tables) for storing and querying data, loading a small amount of sample data from CSV files into a Snowflake table and then querying the table.
3. [Create an S3 bucket](#): `s3-snowflakeintegration-accountId-region`. Replace accountId and region with the AWS Account ID and region of your AWS account.
 - a. [Create a folder](#) in your S3 bucket called `SnowflakeIntegration_Lambda` and upload the [SnowflakeIntegration_Lambda.zip](#) file. This lambda uses the [Snowflake Connector for Python](#) to query and update Snowflake
 - b. Upload the [snowflakelayer.zip](#) in the root folder of this S3 bucket. This zip file packages the Snowflake python connector as an [AWS Lambda Layer](#)
 - c. [Create a folder](#) in your S3 bucket called `template` and upload the [aws-snowflake-integrationobject.yml](#) CloudFormation template. This template is provisioned when the Service Catalog Product is launched and it automates this integration for Snowflake to access S3

Deployment and Configuration Steps

Step 1: Install

The solution is installed in a single step. Create a stack from the [AWS CloudFormation console](#) by launching the [aws-snowflakeintobj-servicecatalog](#) template. The template takes the following parameter:

- **S3StagingBucketURL** - This is the HTTP URL of the S3 bucket (`s3-snowflakeintegration-accountId-region`) that you configured in Step 2 in the prerequisites section.

Step 2: Test and Run

1. The Snowflake solution creates a Snowflake [Service Catalog Portfolio](#), a 'SnowflakeEnduserGroup' [AWS IAM](#) group and provides this IAM group with access to the Portfolio. In order to launch the Snowflake Service Catalog Product, you have 2 options –
 - a. Option 1 - [Follow the steps here to grant your current logged in AWS IAM user/role permissions to access the Service Catalog Portfolio](#) and launch the Snowflake Service Catalog product using your current logged in IAM user/role.
 - b. Option 2 – [Add an IAM user](#) to the 'SnowflakeEnduserGroup' IAM group. [Log in](#) as this IAM user to launch the Snowflake Service Catalog Product
2. [Navigate to the Service Catalog Console](#) in your AWS account. [Follow instructions here to launch your Service Catalog Product](#) – 'SnowflakeStorageIntegrationProduct'
 - a. Provide Snowflake connection details and the external S3 bucket name and S3 prefix as parameters
3. [Navigate to the AWS IAM console](#) of your AWS account and check that a new IAM role has been provisioned that ends with S3INTxxxxx suffix. This suffix will also be the name of your new Snowflake integration object
4. [Log into your Snowflake web interface](#) and ensure that the URL contains your [account identifier](#) (or [use snowsql as outlined here](#))
 - a. Validate that a new Snowflake integration object has been created (DESC INTEGRATION 'integrationobjectname')
 - b. Obtain the AWS_IAM_USER_ARN and AWS_EXTERNAL_ID parameters from 3a in the preceding step. [Navigate to the IAM console](#) of your AWS account and check that the AWS IAM role in Step 2 uses those parameters in its trust policy as the principal and external id parameters respectively
 - c. Validate that a new storage object has been created in Snowflake that references the S3 bucket and uses the integration object (SHOW STAGES IN ACCOUNT)

Solution Estimated Pricing

Contact [Snowflake team](#) to learn more.

FAQs

You can find a list of FAQs and documentation from our [Snowflake Community](#)

Additional resources

- [Snowflake on AWS](#)

Partner contact information

For general inquiries, please refer to the [Snowflake Partner Portal](#)