



Architecture Diagrams

Protegrity Data Protection for Amazon S3 and Snowflake



Protegrity Data Protection for Amazon S3 and Snowflake: Architecture Diagrams

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

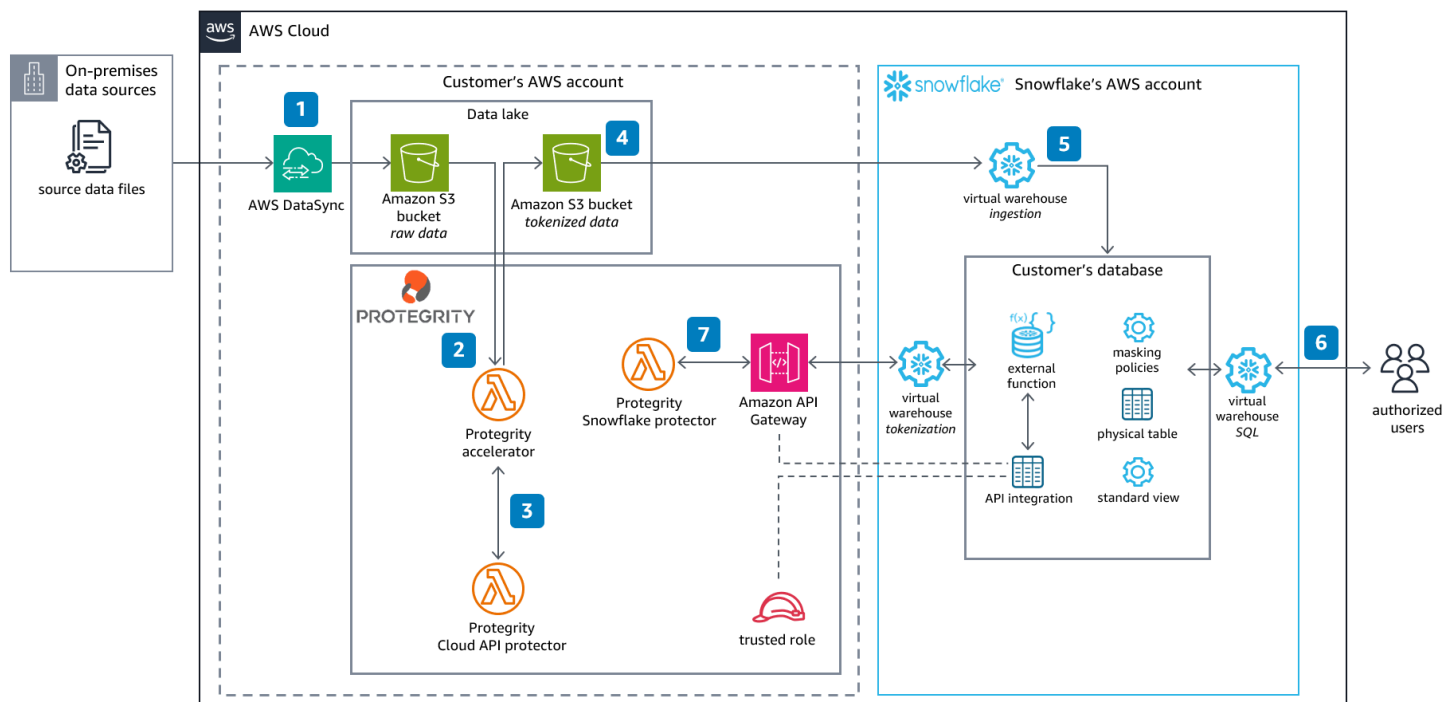
Home	i
Protegrity Data Protection for Amazon S3 and Snowflake Diagram	1
Download editable diagram	2
Create a free AWS account	2
Further reading	2
Contributors	3
Diagram history	3

Protegrity Data Protection for Amazon S3 and Snowflake

Publication date: **October 12, 2023** ([Diagram history](#))

This architecture shows how Protegrity on AWS can be used to protect sensitive data in Amazon S3 and then show the same data as clear text based on permissions from Snowflake.

Protegrity Data Protection for Amazon S3 and Snowflake Diagram



1. External Files are sent to an **Amazon S3** (Amazon S3) input bucket by **AWS DataSync**.
2. The **Amazon S3** Protegrity accelerator that was built using **AWS Lambda** is initiated by an **Amazon S3** event. The accelerator reads the data from the **Amazon S3** bucket and invokes the Protegrity Cloud API protector.
3. Protegrity Cloud API protector, which was built using **Lambda**, applies data protection on the data. The Protegrity Cloud API protector returns protected (encrypted or tokenized) data if the passed user has the right permissions.
4. The **Amazon S3** Protegrity accelerator receives the protected data and creates a new object in the output **Amazon S3** bucket (data lake). Optionally, data is deleted from the raw data bucket.

5. Data from the **Amazon S3** data lake is loaded into a Snowflake table by a Snowflake virtual warehouse. A masking policy is applied on that table.
6. When a user queries a dataset containing protected data, Snowflake's masking policy invokes the Protegrity Snowflake protector by using an external function. This process is managed by a Snowflake virtual warehouse. It's worth noting the distinct workload isolation and immediate scaling capability of Snowflake, as demonstrated in steps 5 and 6, through independently scalable virtual warehouses.
7. The Snowflake external function call goes through **Amazon API Gateway**. The authorization of this service is achieved using Snowflake's API integration object, which encompasses **API Gateway** and trusted roles created for REST API egress from Snowflake's **Amazon Virtual Private Cloud** (Amazon VPC) to the customer's AWS account. The Protegrity protector returns clear text data for users with the right permissions.

Download editable diagram

To customize this reference architecture diagram based on your business needs, [download the ZIP file](#) which contains an editable PowerPoint.

Create a free AWS account

[Sign up now](#)

Sign up for an AWS account. New accounts include 12 months of [AWS Free Tier](#) access, including the use of Amazon EC2, Amazon S3, and Amazon DynamoDB.

Further reading

For additional information, refer to

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)
- [Protegrity](#)
- [Snowflake](#)

Contributors

Contributors to this reference architecture diagram include:

- Venkatesh Aravamudan, Partner Solutions Architect, Amazon Web Services
- Bosco Albuquerque, Senior Partner Solutions Architect, Amazon Web Services
- Tamara Astakhova, Senior Partner Solutions Architect, Amazon Web Services

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	October 12, 2023

Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.