Enrollment Data Cloud Solution

09/19/2024

Contents

[Architecture: 2](#_Toc177632281)

[Tools and Technologies: 2](#_Toc177632282)

[Why this model: 3](#_Toc177632283)

[Future Improvements: 3](#_Toc177632284)

## Architecture:

A diagram of a software company

Description automatically generated

* Develop code in Python to ping API to retrieve Enrollment data for 2021 for Grade-Pk.
* Save the return data as csv file to local folder.
* Access S3 bucket via boto3 to push all csv file in the given folder.
* Schedule API ping, creating csv and pushing them to S3.
* Setup SQS in AWS to notify when a new file arrives in S3 bucket.
* Create a database to ingest the data.
* Setup security integration, stage and pipe for ingestion.
* Query the data

## Tools and Technologies:

1. VS Code
   1. Python
   2. Boto3
   3. GIT
2. AWS
   1. IAM
      1. User
      2. Role
      3. Policies
   2. S3
   3. SQS
3. Snowflake
   1. Storage Integration
   2. Stage
   3. Pipe

## Why this model:

There are many ways to develop a ETL pipeline. This architecture with AWS and Snowflake combined is a powerful platform. Things to consider when developing a cloud service are

1. Scalability
2. Security
3. Availability
4. Maintainability
5. Performance Optimization
6. Cost Efficient

### Scalability

This system is scalable. S3 storage bucket is not restricted to storage limit. Unlimited files can be uploaded. The S3 infrastructure automatically scale to handle high amount of data.

Same goes for Snowflake where it can be scaled both horizontally and vertically. If high compute power or storage is required, it is easily scalable. Snowflake dynamically scales up and down based on the workload.

## Security

S3 buckets can block all the content from public and be restricted to a specific role or user. Bucket policies can be applied to restrict access to objects as well. S3 data is encrypted at rest at the same time encrypted data can also be uploaded. VPC endpoint can also restrict specific AWS services to connect to S3 only.

Snowflake data is also encrypted at rest. Role base access control (RBAC) allow to secure data up to row and column level granularity, this includes data masking. Multi factor authentication can be implemented to authenticate users.

### Availability

AWS environments can be setup with a fallback region. S3 offer to store data across multiple AZ. Other AWS resources if use such as EC2 or RDS provides automatic failover.

Backbone of Snowflake can be on any cloud solution such as AWS, Azure or Google. Users can benefit from which cloud platform to choose also cross-cloud platform replication. Snowflake automatically replicate data in multiple zones for that data to be always available.

Snowflake claims to perform updates without service interruptions. Time travel a powerful feature allows user to access historical data.

### Maintainability

S3 and IAM requires minimum maintenance, that is to review policies, access keys rotations, permissions and such.

Snowflake is highly maintainable. No hardware or infrastructure administration is needed as Snowflake itself is a fully managed service.

### Performance Optimization

AWS services are designed to be performance efficient. S3 offers different storage class for different types of access frequencies, such as Standard or Glacier. It also offered transfer acceleration to improve upload and download speeds.

Snowflake is big on performance optimization. Virtual warehouses for compute are auto scalable to provide high performance during peak times. Query caching for frequently accessed queries results in high performance. Resource management and data sharing with different region to minimize overhead and much more that Snowflake offers to optimize performance.

### Cost Efficient

Both S3 and Snowflake needs to be regularly analyzed for usage patterns, storage needs and querying. Although the storage cost for both is nominal but adds up given the usage and frequency. Need to be monitored to keep the system cost efficient.

## Future Improvements:

1. Run Python via Lambda to ping API
2. Directly store from Lambda to S3
3. Schedule events to execute Lambda function
4. Setup SES for email notification, for failures in real time.
5. Setup email notification after data is loaded into the table, once a day.
6. Further secure the S3 bucket.