8/22/2023

Integrate NICE CXone Data with Snowflake or Other Analytics Store

API integration using Azure ADF & Snowflake

Atanu Das
SENIOR, EY GDS

Contents

1. Introduction:

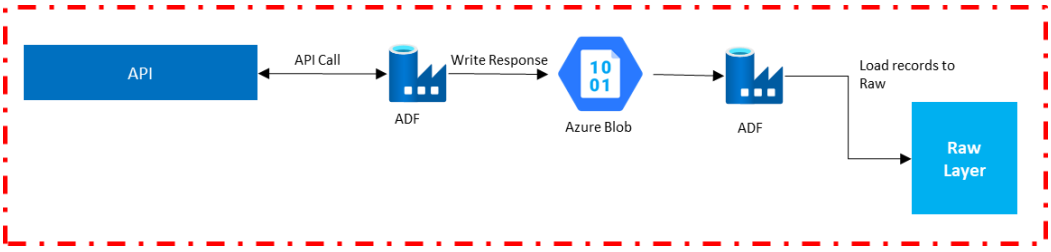
NICE CXone is a cutting-edge cloud-based customer experience platform that empowers organizations to deliver exceptional customer interactions. With a comprehensive suite of Contact Centre tools, CXone offers seamless omnichannel communication, workforce optimization, analytics, and AI-driven insights, enhancing agent productivity and customer satisfaction.
This platform provides different APIs to access the data stored within it for any external reporting or data analysis. The APIs are categorized depending on the nature of the call & functionality as below:

- Admin API: A single API call for all admin-related data like agent details, team, and skill details without any parameter
- Reporting API (#1): This takes the date range as an input parameter and returns all data for the given range like completed contacts, a summary of agents etc.
- Reporting API (#2): This is 3rd-level data when the input is a single agent ID or contact ID and the output is details about the same.

We will discuss the call process in the below sections.

Each HTTPS request requires a valid "API access token," which is retrieved via an OAuth2, Access Key or OpenID Connect authentication process. Because the RESTful API works over HTTPS, it can be used from any platform, programming language, or environment that supports HTTPS.

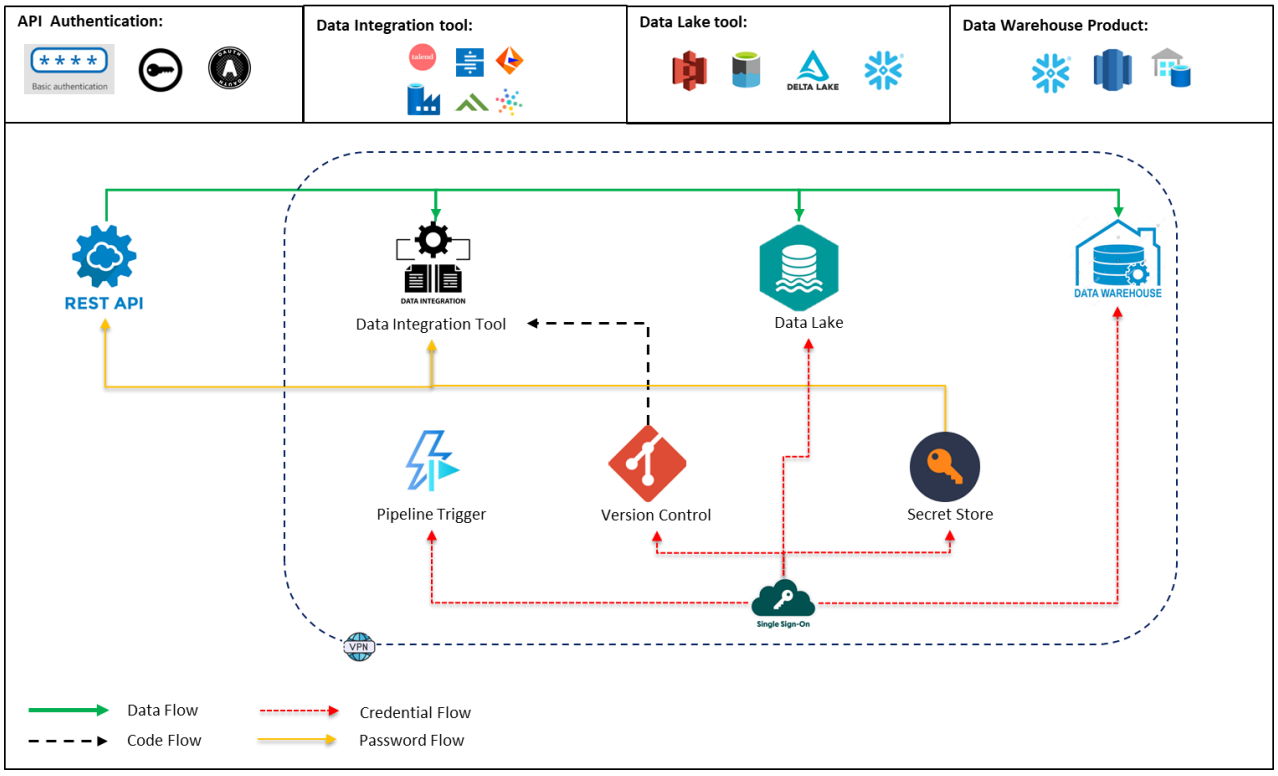
In this use case, we will explore how the APIs will be accessed using ADF & retrieved data to be stored in Snowflake.



The official website for the NICE CXone developer portal: Home (niceincontact.com)

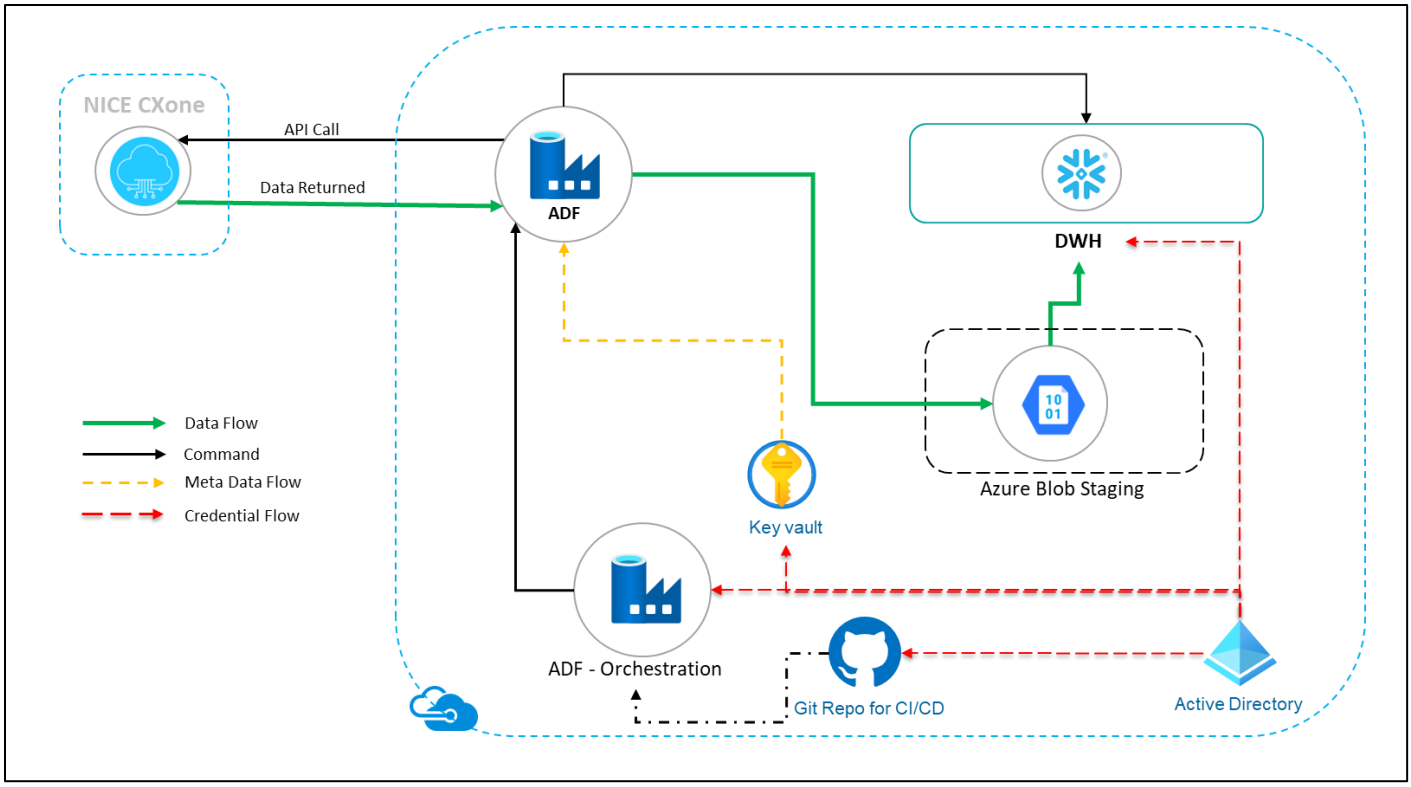
2. HDL Architecture:

The diagram below illustrates the HDL architecture of the process:



3. Data Solution Architecture:

The diagram below illustrates the data solution architecture using Azure, Snowflake & GitHub for the process:

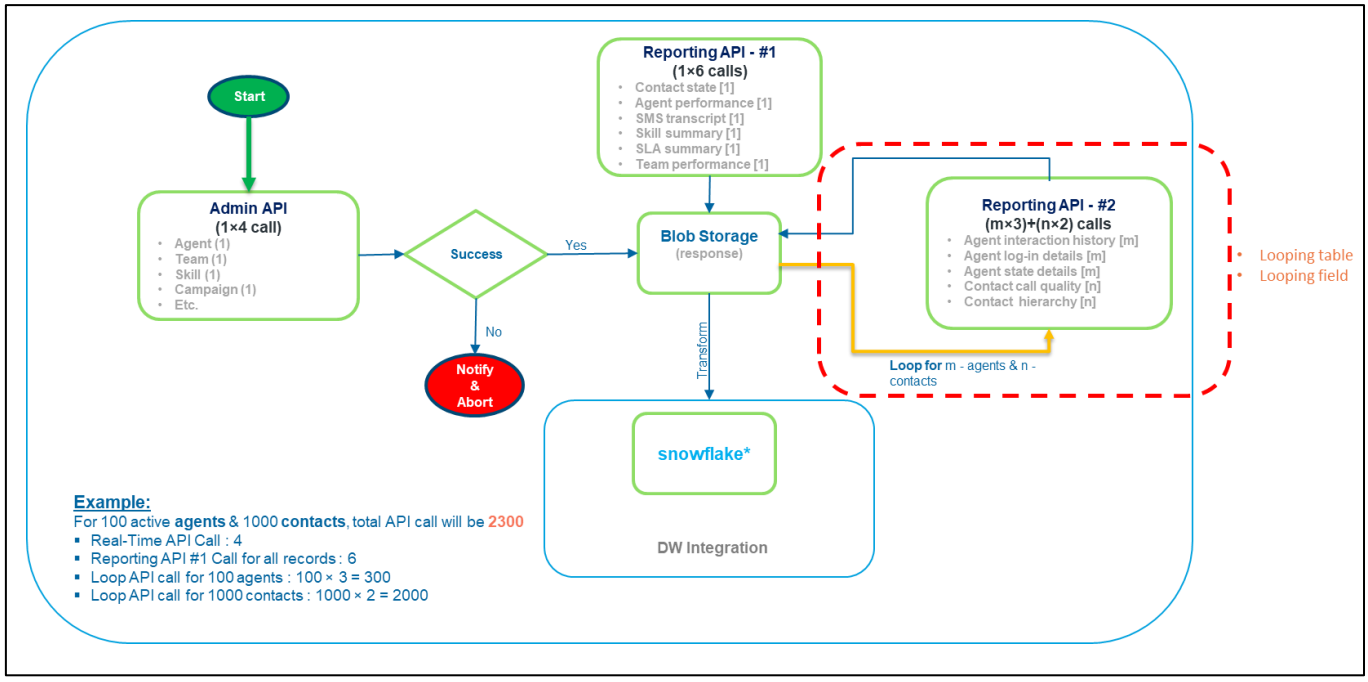

NICE CXone
API Call
Data Returned
ADF
DWH
Azure Blob Staging
Key vault
ADF - Orchestration
Git Repo for CI/CD
Active Directory
Data Flow
Command
Meta Data Flow
Credential Flow

4. Components and Configuration:

Components	Category	Purpose
NICE API	Data Source	Retrieve data stored in the NICE environment
Azure Data Factory	Data Integration & Orchestration Tool	
Snowflake	Cloud Data Store	Data Store for API output
Azure Blob Store	Data Store	Temporary Data Store for API JSON output
Azure Key Vault	Secret Store	Secret store for API keys, passwords etc.
GitHub	Version control tool	Version control for CI/CD
Azure Active Directory	Single Sign On	

5. Workflow:

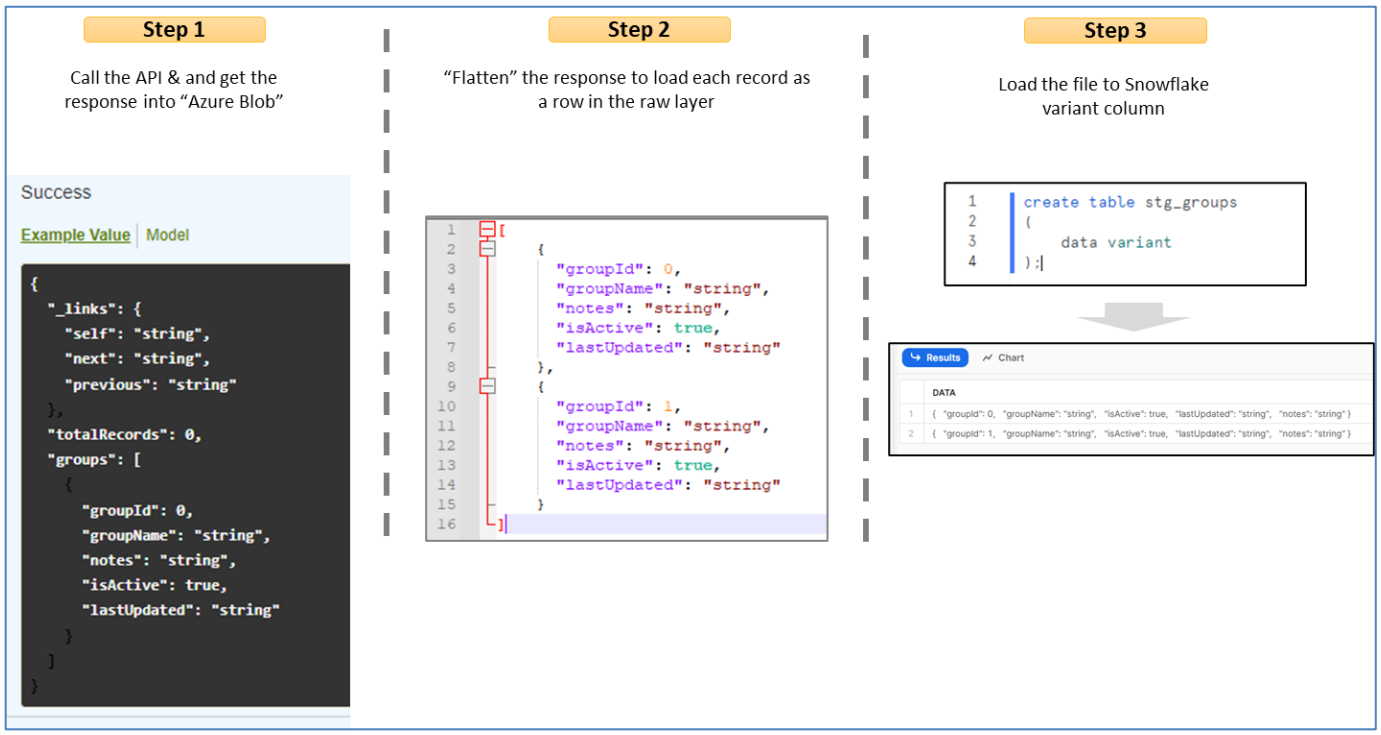
The following diagram illustrates the process of calling reporting & and real-time data APIs and loading data into Snowflake



6. Data Transformation (if applicable):

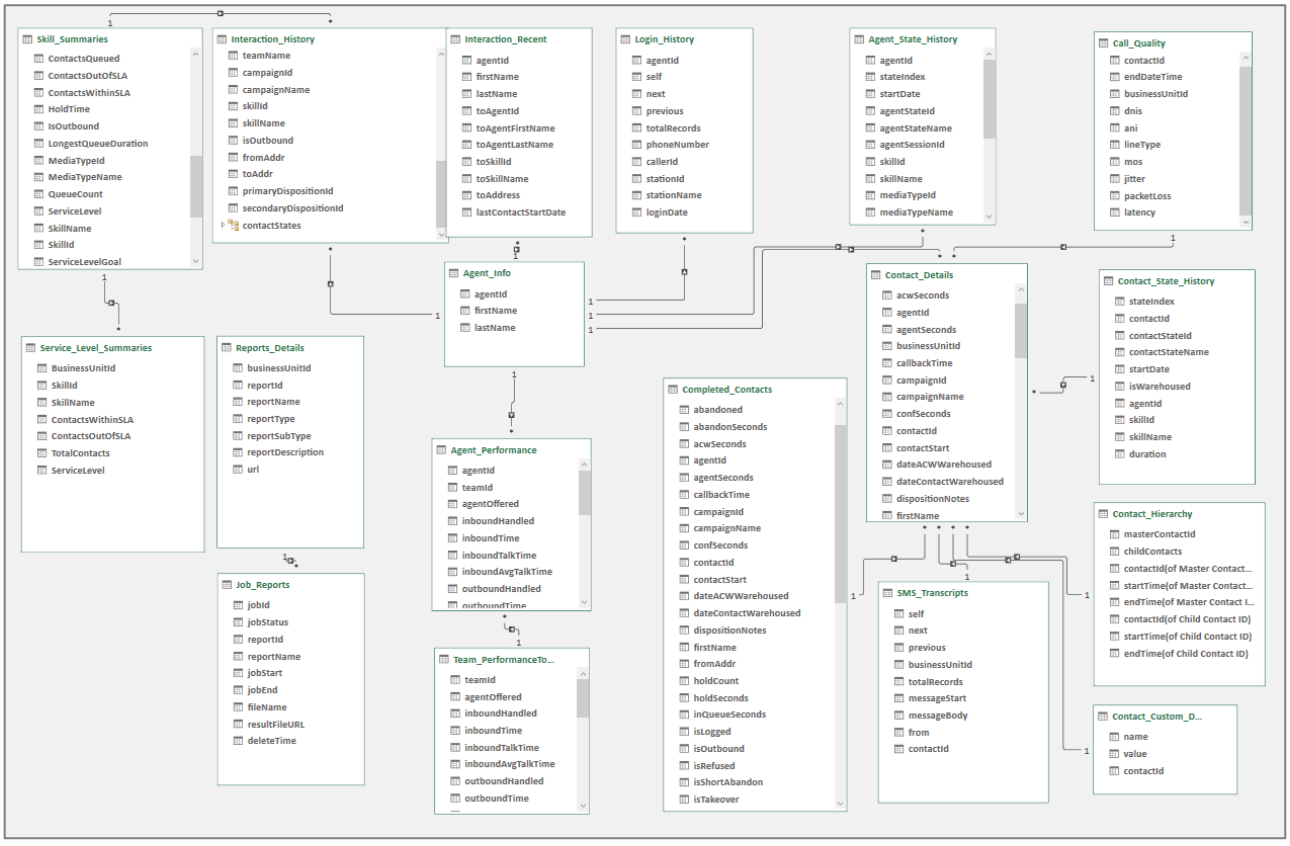
As all the APIs return a single JSON object as output, we need to flatten it to multiple JSON objects before loading it into the raw layer. Once the JSON object for multiple records is available in the Snowflake raw layer, we need to flatten that to tabular format.

Step 1: Get the single object using an API endpoint
Step 2: Create multiple objects for multiple records
Step 3: Load each JSON object to a variant column in the snowflake raw layer.
Step 4: Create structured tabular data from the JSON variant column.



7. Captured Data – Sample Data Models

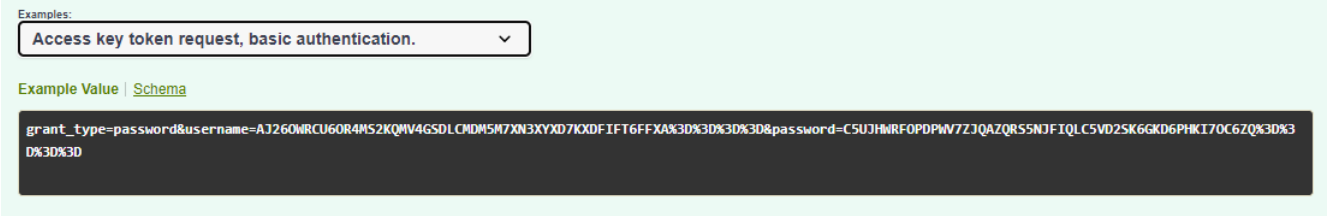
Below Data Model is build based on the APIs output schema mentioned in below here (API (niceincontact.com) and only for example purpose.



8. Security and Authentication:

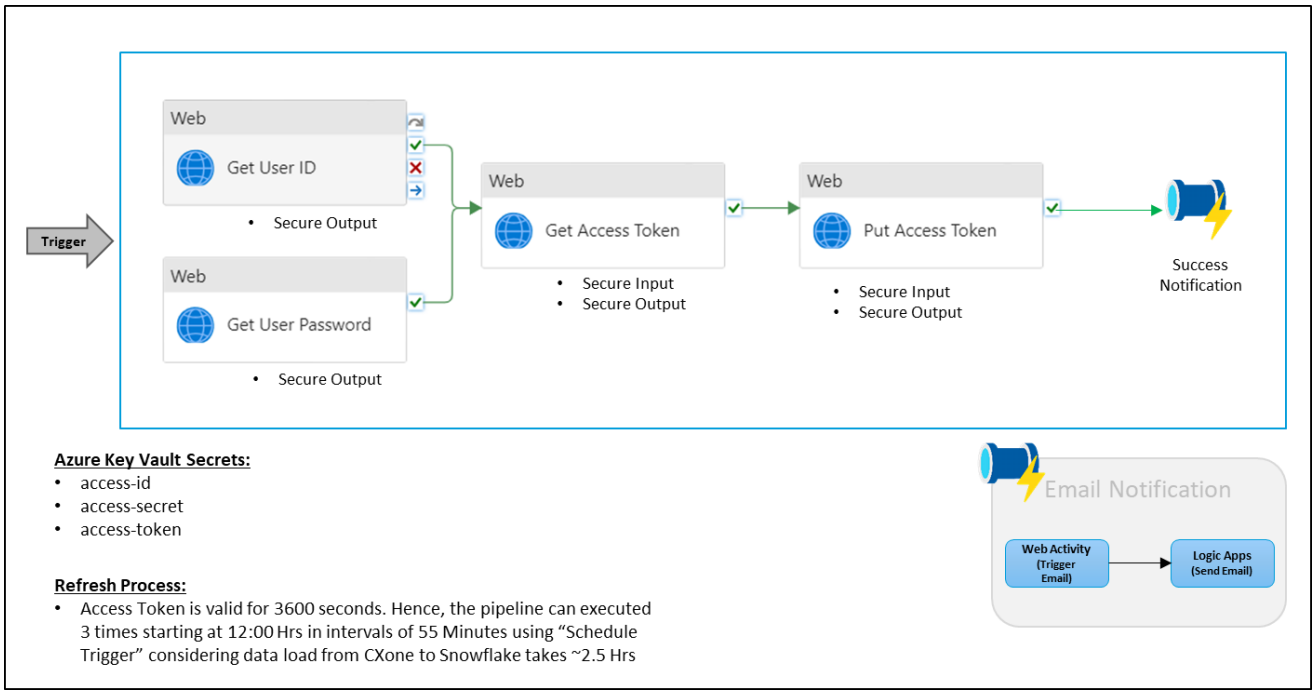
Each API call requires a valid "API access token," which is retrieved via an OAuth2, Access Key or OpenID Connect authentication process. Because the RESTful API works over HTTPS, it can be used from any platform, programming language, or environment that supports HTTPS.

API call to retrieve access-token:



API access token once retrieved using access keys is valid for 60 minutes only. We can have an ADF pipeline as explained below:

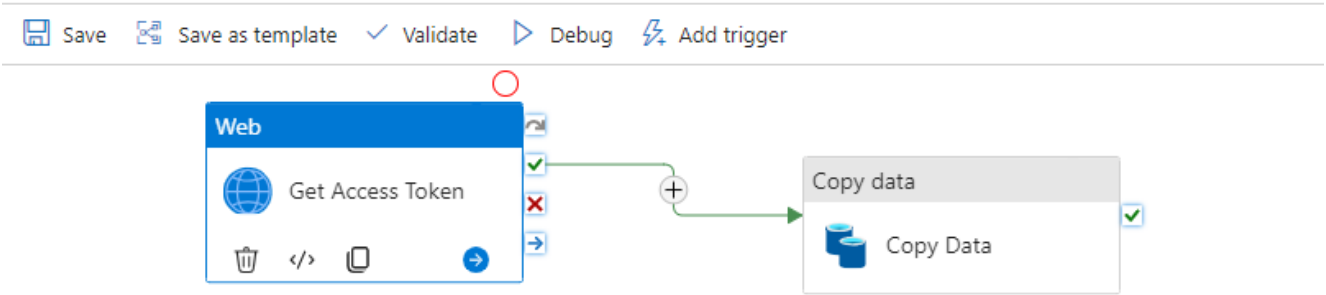
Details here: [Authentication API (niceincontact.com)]



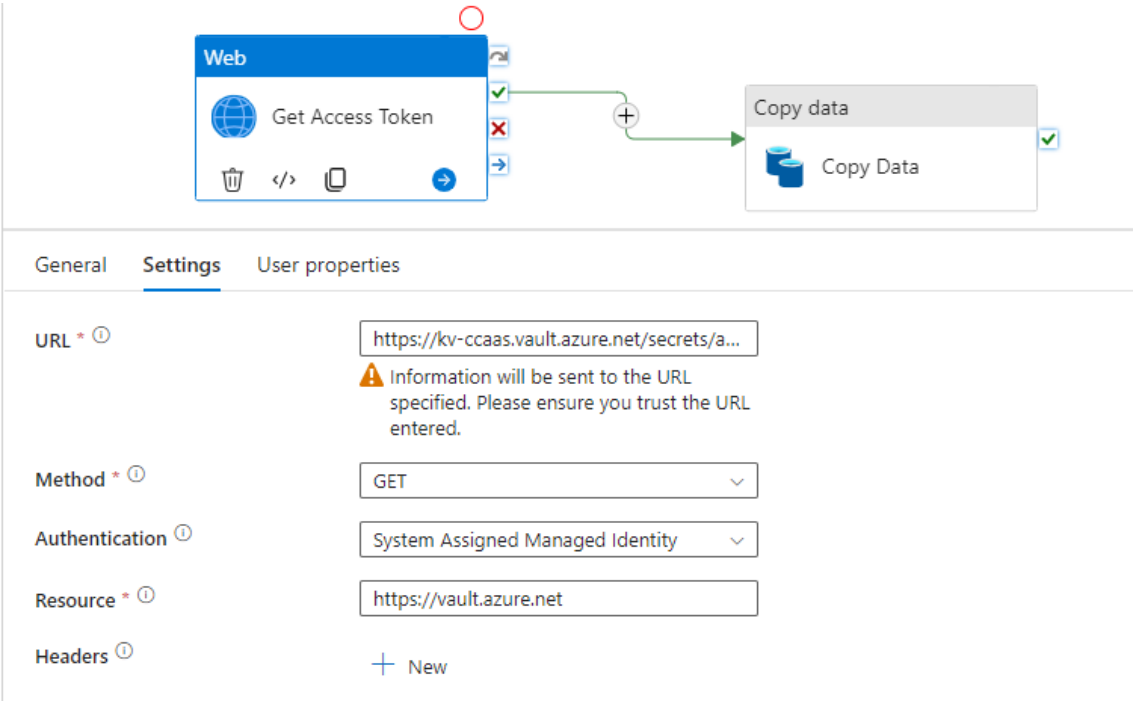
The access token is refreshed in a particular time interval and updated in the Azure key vault.

9. ADF Pipeline Setup:

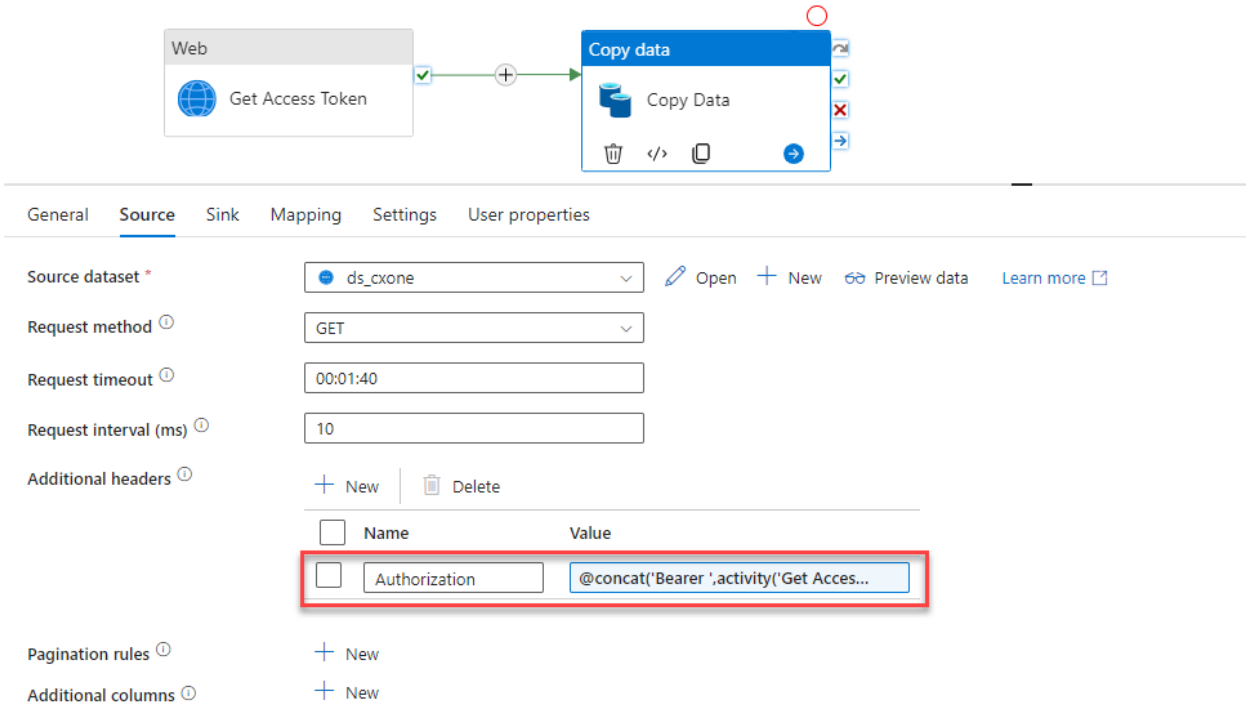
An ADF pipeline with a single web & copy activity to retrieve the data and load the same into the Snowflake variant column. However, flatting the JSOS and loading the tabular data into another table was done within the snowflake task.



Web Activity: This activity retrieves the access token from the Azure key vault that was created and stored in the previous step.



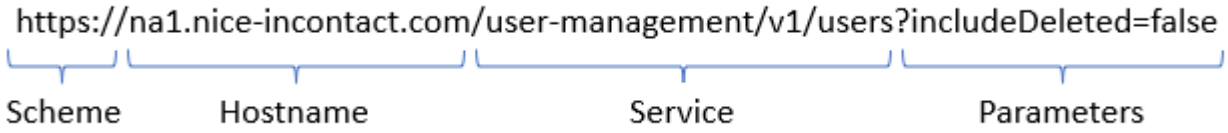
Copy Activity: This activity read API using an API connector and loads the JSON data into Snowflake using Azure blob store as the staging area.



However, the end pipeline can be a little complex considering we have different types of API.

10. Dynamic Execution of API:

This URL consists of several parts, some fixed and others dependent on the user making the call. The following diagram defines these parts.

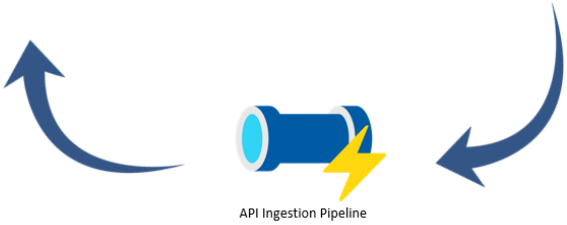


In simple language – the API URL consists of 3 parts; **base_url/relative_url?parameters**. Base URL is fixed for a system, but relative URL & and parameters change for each API.

Let's created a config table to store all the relative URL & parameter required for the APIs and used the same to call a parameterized dynamic copy pipeline.
The below diagram is an example of "agent" and" contact" API and its entry to config table.

config.pipeline_configuration_parameter

Object Name	Object ID	Stage Name	Stage ID	Parameter Name	Parameter Value
agent	1	Raw	1	relative url	agents
agent	1	Raw	1	unrollBy	agents
agent	1	Raw	1	targetTable	agent
agent	1	Raw	1	blobContainer	cxone-api
contact	2	Raw	1	relative url	contacts/completed
contact	2	Raw	1	unrollBy	completedContacts
contact	2	Raw	1	targetTable	contact_detail
contact	2	Raw	1	blobContainer	cxone-api
contact	2	Raw	1	requestParam	{ "Start":"2020-02-11", "end":2020-02-12" }



- New API == An entry into the object & param table
- No need to maintain multiple ingestion pipeline for API

- Development time reduced by 70%
- Maintenance effort reduced by 80%

11. Scalability and Performance:

ADF is a serverless integration and orchestration tool that can be scaled when required. Its Azure integration runtime (Azure IR) compute engine can be vertically or horizontally scaled to handle the increasing data volume.

12. Conclusion:

NICE CXone is a cutting-edge cloud-native solution for customer contact center. The ability to integrate its underlying with modern-age analytics products like Snowflake and Databricks gives the customer an edge to understand customer sentiments, business issues and many more insights that help the organization make a uniform data-driven decision.