Hybrid ETL with existing on-premises SSIS and Azure Data Factory

09/20/2018 • 5 minutes to read • Contributors all

In this article

Organizations that migrate their SQL Server databases to the cloud can realize tremendous cost savings, performance gains, added flexibility, and greater scalability. However, reworking existing extract, transform, and load (ETL) processes built with SQL Server Integration Services (SSIS) can be a migration roadblock. In other cases, the data load process requires complex logic and/or specific data tool components that are not yet supported by Azure Data Factory v2. Commonly used SSIS capabilities include Fuzzy Lookup and Fuzzy Grouping transformations, Change Data Capture (CDC), Slowly Changing Dimensions (SCD), and Data Quality Services (DQS).

To facilitate a "lift and shift" migration of an existing SQL database, a hybrid ETL approach may be the most suitable option. A hybrid approach uses Data Factory as the primary orchestration engine, but continues to leverage existing SSIS packages to clean data and work with on-premises resources. This approach uses the Data Factory SQL Server Integrated Runtime (IR) to enable a "lift and shift" migration of existing databases into the cloud, while using existing code and SSIS packages.

This example scenario is relevant to organizations that are moving databases to the cloud and are considering using Data Factory as their primary cloud-based ETL engine while incorporating existing SSIS packages into their new cloud data workflow. Many organizations have significant invested in developing SSIS ETL packages for specific data tasks. Rewriting these packages can be daunting. Also, many existing code packages have dependencies on local resources, preventing migration to the cloud.

Data Factory lets customers take advantage of their existing ETL packages while limiting further investment in on-premises ETL development. This example discusses potential use cases for leveraging existing SSIS packages as part of a new cloud data workflow using Azure Data Factory v2.

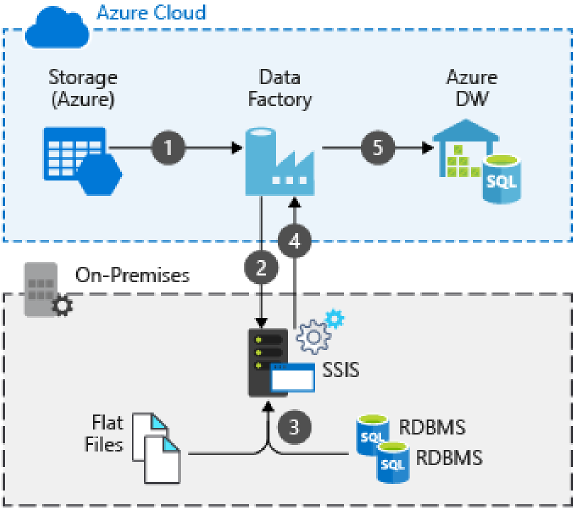
Potential use cases

Traditionally, SSIS has been the ETL tool of choice for many SQL Server data professionals for data transformation and loading. Sometimes, specific SSIS features or third-party plugging components have been used to accelerate the development effort. Replacement or redevelopment of these packages may not be an option, which prevents customers from migrating their databases to the cloud. Customers are looking for low impact approaches to migrating their existing databases to the cloud and taking advantage of their existing SSIS packages.

Several potential on-premises use cases are listed below:

- Loading network router logs to a database for analysis.
- Preparing human resources employment data for analytical reporting.
- Loading product and sales data into a data warehouse for sales forecasting.
- Automating loading of operational data stores or data warehouses for finance and accounting.

Architecture



1. Data is sourced from Blob storage into Data Factory.
2. The Data Factory pipeline invokes a stored procedure to execute an SSIS job hosted on-premises via the Integrated Runtime.
3. The data cleansing jobs are executed to prepare the data for downstream consumption.
4. Once the data cleansing task completes successfully, a copy task is executed to load the clean data into Azure.
5. The clean data is then loaded into tables in the SQL Data Warehouse.

Components

- Blob storage is used to store files and as a source for Data Factory to retrieve data.
- SQL Server Integration Services contains the on-premises ETL packages used to execute task-specific workloads.
- Azure Data Factory is the cloud orchestration engine that takes data from multiple sources and combines, orchestrates, and loads the data into a data warehouse.
- SQL Data Warehouse centralizes data in the cloud for easy access using standard ANSI SQL queries.

Alternatives

Data Factory could invoke data cleansing procedures implemented using other technologies, such as a Databricks notebook, Python script, or SSIS instance running in a virtual machine. Installing paid or licensed custom components for the Azure-SSIS integration runtime may be a viable alternative to the hybrid approach.

Considerations

The Integrated Runtime (IR) supports two models: self-hosted IR or Azure-hosted IR. You first must decide between these two options. Self-hosting is more cost effective but has more overhead for maintenance and management. For more information, see Self-hosted IR. If you need help determining which IR to use, see Determining which IR to use.

For the Azure-hosted approach, you should decide how much power is required to process your data. The Azure-hosted configuration allows you to select the VM size as part of the configuration steps. To learn more about selecting VM sizes, see VM performance considerations.

The decision is much easier when you already have existing SSIS packages that have on-premises dependencies such as data sources or files that are not accessible from Azure. In this scenario, your only option is the self-hosted IR. This approach provides the most flexibility to leverage the cloud as the orchestration engine, without having to rewrite existing packages.

Ultimately, the intent is to move the processed data into the cloud for further refinement or combining with other data stored in the cloud. As part of the design process, keep track of the number of activities used in the Data Factory pipelines. For more information, see Pipelines and activities in Azure Data Factory.

Pricing

Data Factory is a cost-effective way to orchestrate data movement in the cloud. The cost is based on the several factors.

- Number of pipeline executions
- Number of entities/activities used within the pipeline
- Number of monitoring operations
- Number of Integration Runs (Azure-hosted IR or self-hosted IR)

Data Factory uses consumption-based billing. Therefore, cost is only incurred during pipeline executions and monitoring. The execution of a basic pipeline would cost as little as 50 cents and the monitoring as little as 25 cents. The Azure cost calculator can be used to create a more accurate estimate based on your specific workload.

When running a hybrid ETL workload, you must factor in the cost of the virtual machine used to host your SSIS packages. This cost is based on the size of the VM ranging from a D1v2 (1 core, 3.5 GB RAM, 50 GB Disk) to E64V3 (64 cores, 432 GB RAM, 1600 GB disk). If you need further guidance on selection the appropriate VM size, see VM performance considerations.

Next Steps

- Learn more about Azure Data Factory.
- Get started with Azure Data Factory by following the Step-by-step tutorial.
- Provision the Azure-SSIS Integration Runtime in Azure Data Factory.