Orchestrating Big Data Solutions with Azure Data Factory

Lab 2 – Creating a Pipeline

Overview

In this lab, you will use Azure Data Factory to implement a simple data pipeline that copies data from a file in Azure Blob Storage to a table in Azure SQL Database.

What You'll Need

To complete the labs, you will need the following:

- A web browser
- A Microsoft account
- A Microsoft Azure subscription
- A Windows, Linux, or Mac OS X computer
- The lab files for this course
- The Azure resources created in the previous lab.

Important: Before starting this lab, make sure you have completed Lab 1.

Exercise 1: Create a Pipeline

You can create custom data flows by defining the linked services, datasets, and pipelines required to transfer and transform your data. In this exercise, you will create a simple pipeline that performs the same copy data task you accomplished with the Copy Data wizard in Lab 1.

Delete the copied transaction data

To avoid confusion, you will delete the database records inserted by the wizard in the previous lab, before recreating the pipeline manually.

1. In the Azure portal, click **All Resources**, and then click your Azure SQL Database.
2. On the database blade, view the **Data Explorer** page. This opens the web-based query interface for your Azure SQL Database.
3. In the toolbar for the query editor, click **Login**, and then log into your database using SQL Server authentication and entering the login name and password you specified when provisioning the Azure SQL Database server.
4. Enter the following command, and then click **Run**:

```
TRUNCATE TABLE dbo.transactions;
```

5. Run the following command to verify that the table is empty again:

   ```
   SELECT * FROM dbo.transactions;
   ```

View Existing Linked Services

Azure Data Factory uses *linked services* to access, transform, and store data. In the case of your pipeline, you require linked services for the blob store account where the source data is stored, and the Azure SQL Database containing the table you want to load.

1. In the Microsoft Azure portal, browse to the blade for your data factory, and click the **Author and deploy** tile.
2. In the pane on the left, expand **Linked Services** and note that linked services named **blob-store** and **sql-database** are already defined for the blob store and SQL database – these were created by the **Copy Data** wizard in the previous lab.
3. Click the **blob-store** linked service to view its JSON definition, which should look like this:

   ```
   {
    "name": "blob-store",
    "properties": {
     "hubName": "adf_name_hub",
     "type": "AzureStorage",
     "typeProperties": {
     "connectionString":
   "DefaultEndpointsProtocol=https;AccountName=your_store;AccountKey=***"
     }
    }
   }
   ```

4. Click the **sql-database** linked service to view its JSON definition, which should look like this:

   ```
   {
    "name": "sql-database",
    "properties": {
     "hubName": "adf_name_hub",
     "type": "AzureSqlDatabase",
     "typeProperties": {
     "connectionString": "Data
   Source=your_server.database.windows.net;Initial
   Catalog=DataDB;Integrated Security=False;User
   ID=SQLUser;Password=**********;Connect Timeout=30;Encrypt=True"
     }
    }
   }
   ```

Create Datasets

Datasets define schema for data flows, including data to be extracted from a source and data to be loaded into a destination (or *sink*). You must create datasets to define the text file data source and the database table to be used in your pipeline.

1. In the **More** menu, click **New dataset**, and then click **Azure Blob storage** to create a new JSON document for an Azure Blob store dataset.

2. In the new JSON document, replace the default code with the following code, which you can copy and paste from **transactions-txt.json** in the folder where you extracted the lab files:

```
{
  "name": "transactions-txt",
  "properties": {
    "structure": [
      {
        "name": "tdate",
        "type": "Datetime"
      },
      {
        "name": "amount",
        "type": "Decimal"
      }
    ],
    "type": "AzureBlob",
    "linkedServiceName": "blob-store",
    "typeProperties": {
      "folderPath": "adf-data/data/",
      "format": {
        "type": "TextFormat",
        "columnDelimiter": ",",
        "firstRowAsHeader": true
      }
    },
    "external": true,
    "availability": {
      "frequency": "Minute",
      "interval": 15
    }
  }
}
```

This JSON defines a schema for comma-delimited text containing a **tdate** column of **DateTime** values, and an **amount** column of **Decimal** values. The data is available in the **adf-data/data** folder of the blob storage account defined by the **blob-store** linked service, and new data will be available every fifteen minutes.

3. Click **Deploy** to deploy the dataset definition to your Azure Data Factory.
4. In the **More** menu, click **New dataset**, and then click **Azure SQL** to create a new JSON document for an Azure SQL Database dataset.
5. In the new JSON document, replace the default code with the following code, which you can copy and paste from **dbo-transactions.json** in the folder where you extracted the lab files:

```
{
  "name": "dbo-transactions",
  "properties": {
    "type": "AzureSqlTable",
    "linkedServiceName": "sql-database",
    "structure": [
      {
        "name": "tdate",
        "type": "Datetime"
      },
      {
        "name": "amount",
        "type": "Decimal"
      }
    ],
    "typeProperties": {
      "tableName": "dbo.transactions"
    },
    "availability": {
      "frequency": "Minute",
      "interval": 15
    }
  }
}
```

This JSON defines a schema for a table named **dbo.transactions** containing a **tdate** column of **DateTime** values, and an **amount** column of **Decimal** values in the database defined by the **sql-database** linked service. The dataset can be updated every fifteen minutes.

6. Click **Deploy** to deploy the dataset definition to your Azure Data Factory.
7. In the pane on the left, expand the **Datasets** folder and verify that the **transactions-txt** and **dbo-transactions** datasets are listed.

Create a Pipeline

Now that you have defined the linked services and datasets for your data flow, you can create a pipeline to encapsulate it. A pipeline defines a series of actions that use linked services to operate on datasets. In this case, your pipeline will consist of a single action to copy data from the **blob-store** linked service to the **sql-database** linked service.

1. In the **More** menu, click **New pipeline** to create a new JSON document for a pipeline.
2. In the new JSON document, replace the default code with the following code, which you can copy and paste from **copytrans.json** in the folder where you extracted the lab files:

```
{
  "name": "copytrans",
  "properties": {
    "activities": [
      {
        "type": "Copy",
        "typeProperties": {
          "source": {
            "type": "BlobSource",
            "recursive": false
          },
          "sink": {
            "type": "SqlSink",
            "writeBatchSize": 0,
            "writeBatchTimeout": "00:00:00"
          },
          "translator": {
            "type": "TabularTranslator",
            "columnMappings": "tdate:tdate,amount:amount"
          }
        },
        "inputs": [
          {
            "name": "transactions-txt"
          }
        ],
        "outputs": [
          {
            "name": "dbo-transactions"
          }
        ],
        "name": "CopyTransData"
      }
    ],
    "pipelineMode": "OneTime"
  }
}
```

This JSON defines a pipeline that includes a **Copy** action to copy the **transaction-txt** dataset in the blob store to the **dbo.transactions** table in the SQL database.

3. Click **Deploy** to deploy the pipeline to your Azure Data Factory.

View Pipeline Status

Now that you have deployed your pipeline, you can use the Azure portal to monitor its status.

1. After the pipeline has been deployed, return to the blade for your Azure Data Factory, and click for the **Pipelines** tile - it may already indicate the **Wizard Copy** pipeline that you created with the **Copy Data** wizard previously – wait for it to also display the **copytrans** pipeline with a **Pipeline State** of **Running**.
2. Click the **copytrans** pipeline, and observe the state of the activities it contains – it may initially contain no activities, as the datasets are validated and the pipeline is prepared.
3. Close the **Pipelines** blade, and wait for around 15 minutes. Then reopen the **Pipelines** blade and the **copytrans** blade to view its state. When the pipeline has been run, its **State** in the **copytrans** blade will be indicated as **Ready**, and in the **Pipelines** blade its **Pipeline State** will be set to **Completed**.

Verify that the Data Has Been Copied

Now that your pipeline has run, you can verify that the data has been copied to your SQL database.

1. Return to the Query editor for your Azure SQL Database and run the following query:

   ```
   SELECT * FROM dbo.transactions;
   ```

2. Verify that the table now contains transaction data, copied from the text file in your blob store.

Note: You will use the resources you created in this lab when performing the next lab, so do not delete them.