Orchestrating Big Data Solutions with Azure Data Factory

Lab 1 - Getting Started with Azure Data Factory

Overview

In this lab, you will provision an Azure Data Factory, and use the Copy Wizard to copy 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

Note: To set up the required environment for the lab, follow the instructions in the Setup document for this course.

Exercise 1: Provisioning Azure Resources

In this exercise, you will create the Azure Storage account, Azure SQL Database instance, and Azure Data Factory instance.

Note: The Microsoft Azure portal is continually improved in response to customer feedback. The steps in this exercise reflect the user interface of the Microsoft Azure portal at the time of writing, but may not match the latest design of the portal exactly.

Create a Storage Account and a Blob Container

The source data for your data pipeline will be stored in an Azure storage account:

1. In the Microsoft Azure portal, in the menu, click **New**. Then in the **Storage** menu, click **Storage account**.
2. In the **Create storage account** blade, enter the following settings and click **Create**:
    - **Name**: *Enter a unique name (and make a note of it!)*
    - **Deployment model**: Resource manager
    - **Account kind**: General purpose
    - **Performance**: Standard

- **Replication**: Locally-redundant storage (LRS)
- **Storage service encryption**: Disabled
- **Subscription**: *Select your Azure subscription*
- **Resource group**: *Create a new resource group with a unique name*
- **Location**: *Select any available region*

3. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the storage account to be deployed (this can take a few minutes.)
4. After the storage account has been created, browse to its blade in the Azure portal.
5. On the blade for your storage account, click **Blobs**, and add a container with the following properties:
    - **Name**: adf-data
    - **Access type**: Private
6. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the container to be created (this should take a few seconds.)
7. After the container has been created, return to the blade for your storage account, and click **Access keys**. Note that this blade lists the storage account name and two keys that client applications can use for authentication when connecting.

Create an Azure SQL Database

Your data pipeline will copy the source data to an Azure SQL Database. SQL databases are hosted in servers, so you will create both a database and a server to host it.

1. In the Microsoft Azure portal, in the menu, click **New**. Then in the **Databases** menu, click **SQL Database**.
2. In the **SQL Database** blade, enter the following settings, and then click **Create**:
    - **Name**: DataDB
    - **Subscription**: *Select your Azure subscription*
    - **Resource Group**: *Select the resource group you created previously*
    - **Select source**: Blank database
    - **Server:** *Create a new server with the following settings*:
        - **Server name**: *Enter a unique name (and make a note of it!)*
        - **Server admin login:** *Enter a user name of your choice (and make a note of it!)*
        - **Password:** *Enter and confirm a strong password (and make a note of it!)*
        - **Region:** *Select your HDInsight cluster location*
        - **Create V12 server (Latest update)**: Yes
        - **Allow azure services to access server**: Selected
    - **Elastic pool**: *Not enabled*
    - **Pricing tier:** *View all and select* Basic
    - **Collation**: SQL_Latin1_General_CP1_CI_AS
    - **Pin to dashboard:** Unselected
3. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the SQL database to be deployed (this can take a few minutes.)
4. After the database has been created, browse to your Azure SQL server (not the database) and under **Settings**, click **Properties**.
5. Note the fully qualified name of your server (which should take the form *server*.database.windows.net, where *server* is the server name you specified earlier) and the server admin user name (which should be the login you specified earlier).
6. Under **Settings**, click **Firewall**, and in the **Firewall** blade, note that your client IP address has been automatically detected.
7. Click **Add client IP** to create a rule that permits access to the server from your local computer. Then click **Save** to save the rule. When the firewall rule change is confirmed, click **OK**.

Note: If your client IP address changes, you will need to edit this rule to restore access from your client computer's new address. In some cases, your computer's IP address may be abstracted behind a local firewall or router. If you experience errors when trying to connect to your Azure SQL server in subsequent procedures, try creating a firewall rule that permits access to the IP address range **0.0.0.0** to **255.255.255.255** (this permits access from any Internet-connected device, so you should generally not do this for production servers!). If this still does not resolve the problem, your local firewall may be blocking outbound connections on port 1433 - refer to the documentation for your firewall product to resolve this issue.

For more details about Azure SQL Database firewalls, including troubleshooting tips, see https://azure.microsoft.com/en-us/documentation/articles/sql-database-firewall-configure/.

Create an Azure Data Factory

Now that you have your data stores in place, you are ready to create an Azure Data Factory.

1. In the Microsoft Azure portal, in the menu, click **New**. Then in the **Intelligence + analytics** menu, click **Data Factory**.
2. In the **New data factory** blade, enter the following settings, and then click **Create**:
    - **Name**: *Enter a unique name (and make a note of it!)*
    - **Subscription**: *Select your Azure subscription*
    - **Resource Group**: *Select the resource group you created previously*
    - **Location**: *Select the location you specified for your storage account (if it is not available, select any other location)*
    - **Pin to dashboard:** Unselected
3. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the data factory to be deployed (this can take a few minutes.)

Exercise 2: Using the Azure Data Factory to Copy Data

For simple data copy pipelines, Azure Data Factory provides an easy to use wizard. In this exercise, you will use the wizard to copy data from your Azure blob store account to your Azure SQL Database.

Upload a Data File to the Blob Container

The source data is a comma-delimited text file containing details of sales transactions.

1. In the **data** subfolder of the folder where you extracted the lab files for this course, open the **transactions.txt** file in a text editor.
2. Review the data this file contains, which consist of multiple rows of dates and amounts. Then close the text editor without saving any changes.
3. Start Azure Storage Explorer, and if you are not already signed in, sign into your Azure subscription.
4. Expand your storage account and the **Blob Containers** folder, and then double-click the **adf-data** blob container you created in the previous procedure.
5. In the **Upload** drop-down list, click **Folder**. Then upload the **data** folder (which contains the **transactions.txt** file) as a block blob to the root of the container.

Create a Table in the Database

You will copy the sales transaction data to a table named **transactions**, which contains **id**, **tdate**, and **amount** fields.

Note: The following steps assume that you are using the cross-platform SQL Server command line interface (mssql). You may use an alternative SQL Server client tool if you prefer.

1. Start your SQL Server client tool of choice, and connect to the **DataDB** database on your Azure SQL Database server (*server*.database.windows.net) using the server admin login credentials you specified when creating the Azure SQL database.

   If you are using the cross-platform SQL Server command line interface, open a command line or console, and enter the following (case-sensitive) command, replacing *server* with your Azure SQL Database Server name, *login* with your server admin login name, and *password* with your server login password. Note that on Linux and Mac OS X operating systems, you may need to prefix any special characters (such as **$**) with a **** character (for example, if your password is *Pa$$w0rd*, enter *Pa\$\$w0rd*):

   ```
   mssql -s server.database.windows.net -u login -p password -d DataDB -e
   ```

2. When connected to the database, enter the following Transact-SQL statement to create a table (note that when using *mssql*, commands must be entered on a single line):

   ```
   CREATE TABLE transactions(id int identity, tdate date, amount decimal);
   ```

3. Keep the SQL Server client tool open. You will return to it in a later procedure.

use the Azure Data Factory Copy Wizard to Copy the Data

1. In the Microsoft Azure portal, browse to the blade for your data factory, and click the **Copy data** tile. This opens a new tab in your browser.
2. On the **Properties** page of the Copy Data wizard, enter the following details and then click **Next**:
   - **Task name**: Wizard Copy
   - **Task description**: Copy transactions
   - **Task cadence (or) Task schedule**: Run once now
   - **Expiration time**: 3:00:00:00
3. On the **Source data store** page, on the **Connect to a Data Store** tab, select **Azure Blob Storage**. Then click **Next**.
4. On the **Specify the Azure Blob storage account** page, enter the following details and then click **Next**:
   - **Connection name**: blob-store
   - **Account selection method**: From Azure subscriptions
   - **Azure subscription**: *Select your subscription*
   - **Storage account name**: *Select your storage account*
5. On the **Choose the input file or folder** page, double-click the **adf-data** blob container you created previously, and then select the **data** folder (which contains the **transactions.txt** file). Then click **Choose**, and click **Next**.
6. On the **File format settings** page, wait a few seconds for the data to be read, and then verify the following details, ensuring that the rows of data in the **Preview** section match the table below, and click **Next**:
   - **File format**: text format
   - **Column delimiter**: Comma (,)
   - **Row delimiter**: Carriage return and line feed (\r\n)
   - **Skip line count**: 0
   - **Column names in first data row**: Selected

- **Treat empty column value as null**: Selected
- **Preview**:

tdate	amount
2016-01-01	129.99
2016-01-01	125.49
2016-01-01	99.75
…	

7. On the **Destination data store** page, on the **Connect to a Data Store** tab, select **Azure SQL Database**. Then click **Next**.
8. On the **Specify the Azure SQL database** page, enter the following details and then click **Next**:
   - **Connection name**: sql-database
   - **Server / database selection method**: From Azure subscriptions
   - **Azure subscription**: *Select your subscription*
   - **Server name**: *Select your Azure SQL server*
   - **Database name:** DataDB
   - **User name**: *The server admin login name you specified when creating the database*
   - **Password**: The password for your Azure SQL server admin login
9. On the **Table mapping** page, in the **Destination** list, select **[dbo].[transactions]** and click **Next**.
10. On the **Schema mapping** page, ensure that the following settings are selected, and click **Next**:

Blob path: adf-data/data/	[dbo].[transactions]	Include this column
tdate (DateTime)	tdate (DateTime)	✓
amount (Double)	amount (Decimal)	✓
Repeatability settings:		
Method: None		

11. On the **Performance settings** page, expand **Advanced settings** to review the default values. Then click **Next**.
12. On the **Summary** page, click **Finish**.
13. On the **Deploying** page, wait for the deployment to complete.

Verify that the Data Has Been Copied

The Copy Data wizard should have created a pipeline, and run it to copy the transactions data from your blob store to your Azure SQL Database.

1. In your SQL Server client tool, enter the following query:

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

2. Verify that the table now contains 10 rows of transaction data, copied from the text file in your blob store.
3. Keep the command window open. You will return to it in the next exercise.

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