**DP 200 - Implementing a Data Platform Solution**

**Lab 7 - Orchestrating Data Movement with Azure Data Factory**

**Estimated Time**: 45 minutes

**Pre-requisites**: It is assumed that the case study for this lab has already been read. It is assumed that the content and lab for module 1: Azure for the Data Engineer has also been completed

**Lab files**: The files for this lab are located in the *Allfiles\Labfiles\Starter\DP-200.7* folder.

**Lab overview**

In this module, students will learn how Azure Data factory can be used to orchestrate the data movement from a wide range of data platform technologies. They will be able to explain the capabilities of the technology and be able to set up an end to end data pipeline that ingests data from SQL Database and load the data into SQL Data Warehouse. The student will also demonstrate how to call a compute resource.

**Lab objectives**

After completing this lab, you will be able to:

1. Explain data streams and event processing
2. Data Ingestion with Event Hubs
3. Processing Data with Stream Analytics Jobs

**Scenario**

After performing the initial population of the Data Warehouse into Azure SQL Data Warehouse, the information services department want to automate this process. You have been asked to support the information services department in developing a solution that can automate the movement of data from Azure SQL Database.

Your solution should be able to perform full copy of [SalesLT].[ProductCategory] and [SalesLT].[ProductDescription] transaction table that act as dimension tables of the same name in your Azure SQL Data Warehouse. Furthermore, the solution should also follow best practices of loading into a Massively Parallel Processing (MPP) system using Azure Data Factory as the orchestrator of the data movements.

In addition, the Data Scientists have asked to confirm if Azure Databricks can be called from Azure Data Factory. To that end, you will create a simple proof of concept Data Factory pipeline that calls Azure Databricks as a compute resource.

At the end of this lab, you will have:

1. Explain how Azure Data Factory works
2. Azure Data Factory Components
3. Azure Data Factory and Databricks

**IMPORTANT**: As you go through this lab, make a note of any issue(s) that you have encountered in any provisioning or configuration tasks and log it in the table in the document located at *\Labfiles\DP-200-Issues-Doc.docx*. Document the Lab number, note the technology, Describe the issue, and what was the resolution. Save this document as you will refer back to it in a later module.

**Exercise 1: Explain how Azure Data Factory works**

Estimated Time: 15 minutes

Group exercise

The main task for this exercise are as follows:

1. From the case study and the scenario, identify the data stream ingestion technology for AdventureWorks, and the high-level tasks that you will conduct as a data engineer to complete the social media analysis requirements.
2. The instructor will discuss the findings with the group.

**Task 1: Identify the data requirements and structures of AdventureWorks.**

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab06-Ex01.docx** from the **Allfiles\Labfiles\Starter\DP-200.6** folder.
2. As a group, spend **10 minutes** discussing and listing the data requirements and data structure that your group has identified within the case study document.

**Task 2: Discuss the findings with the Instructor**

1. The instructor will stop the group to discuss the findings.

**Result**: After you completed this exercise, you have created a Microsoft Word document that shows a table of data streaming ingestion and the high-level tasks that you will conduct as a data engineer to complete the social media analysis requirements .

**Exercise 2: Azure Data Factory Components**

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Create a data factory instance.
2. Create an input linked services
3. Define an Input Dataset
4. Create an output linked services
5. Define an Output Dataset
6. Finalize Settings to Optimize for SQL Data Warehouse
7. Monitor the Pipeline execution
8. Confirm the Azure Data Factory components
9. Verify the data output

**Task 1: Create a data factory instance**

1. In Microsoft Edge, go to the Azure portal tab, select **+ Create a resource**, type **factory**, and then select **Data Factory** from the resulting search. Then select **Create**.
2. Create an new Data Factory with the following options, then click **Create**:
   * **Name**: xx-data-factory, where xx are your initials
   * **Subscription**: Your subscription
   * **Resource group**: awrgstudxx
   * **Version**: V2
   * **Location**: select the location closest to you
   * Leave other options to their default settings

**Note**: The creation of the Data Factory takes approximately 1 minute.

**Task 2: Create an input linked services**

1. In the Azure portal, a message is returned to state that the Azure Data Factory installation has completed successfully, click **Go to resource**.
2. In the **xx-data-factory** screen, click on **Author & Monitor**. Another tab opens up to author an Azure Data Factory solution.
3. In the **Get started** page, select the **Copy Data** tile to launch the Copy Data tool:
4. In the **Properties** page, specify **CopyFromSQLToSQLDW** for the Task name field, and select Next:
5. In the **Source data store** page, complete the following steps and then click on **Next**:
   * click **+ Create new connection**:
   * Select **Azure SQL Database** from the gallery, and select **Continue**.
   * In the **New Linked Service (Azure SQL Database)** page, select your server name **sqlservicexx** and database named **DeptDatabasesxx** from the dropdown list, and specify the username and password. Click **Test Connection** connection to validate the settings, and then click **Finish**.

**Task 3: Define an Input Dataset**

1. In the Copy Data Wizard, in the "Select tables from which to copy the data or use a custom query.", under **Existing Tables**, click on the checkbox next to [SalesLT].[ProductCategory] and [SalesLT].[ProductDescription] and click on **Next**.

**Task 4: Create an output linked services**

1. In the **Destination data store** page, complete the following steps and then click on **Next**:
   * click **+ Create new connection**:
   * Select **Azure SQL Data Warehouse** from the gallery, and select **Continue**.
   * In the **New Linked Service (Azure SQL Data Warehouse)** page, select your server name **sqlservicexx** and database named **DWDB** from the dropdown list, and specify the username and password. Click **Test Connection** connection to validate the settings, and then click **Finish**.

**Task 5: Define an Output Dataset**

1. In the Copy Data Wizard, in the "Table mapping", under **Source**, ensure the checkbox next to [SalesLT].[ProductCategory] and [SalesLT].[ProductDescription] and that the destination is the same name. Click on **Next**.
2. In the Copy Data Wizard, in the "Column mapping", read through the **column mappings** which should be a direct copy of columns between the source and destination. Click on **Next**.

**Task 6: Finalize Settings to Optimize for SQL Data Warehouse**

1. In the Copy Data Wizard, in the "Settings", under **Performance settings**, ensure the checkbox next to **Enable staging** is selected.
2. Next to "Staging account linked service, click on **+ New**, In the New Linked Service page, select your storage account **awsastudxx**, and select **Finish**.
3. In the **Advanced settings** section, ensure that **Allow PolyBase** is checked.
4. In the **Advanced settings** section, deselect the **Use type default** option, click on **Next**.
5. In the **Summary** page, review the settings, and select **Next**

**Task 7: Monitor the Pipeline execution**

1. In the Deployment page, select Monitor to **monitor** the pipeline task.
2. Notice that the Monitor tab on the left is automatically selected. The **Actions** column includes links to **view activity** run details and to rerun the pipeline.
3. To view activity runs that are associated with the pipeline run, select the **View Activity Runs** link in the Actions column. To switch back to the pipeline runs view, select the **Pipelines** link at the top. Select **Refresh** to refresh the list.
4. To monitor the execution details for each copy activity, select the **Details** link under Actions in the activity monitoring view. You can monitor details like the volume of data copied from the source to the sink, data throughput, execution steps with corresponding duration, and used configurations:

**Task 8: Confirm the Azure Data Factory components**

1. In the **Author & Monitor** screen, click on the **Author** icon.
2. Confirm in the **Factory Resources** that there is **1 Pipeline** and **2 Datasets** as defined in the Copy Wizard that you have just executed.
3. Close down the Azure Data Factory Tab in Microsoft Edge

**Task 9: Verify the data output**

1. On the windows desktop, click on the **Start**, and type **"SQL Server"** and then click on **Microsoft SQL Server Management Studio 17**
2. In the **Connect to Server** dialog box, fill in the following details
   * Server Name: **sqlservicexx.database.windows.net**
   * Authentication: **SQL Server Authentication**
   * Username: **xxsqladmin**
   * Password: **P@ssw0rd**
3. In the **Connect to Server** dialog box, click **Connect**
4. Expand the database **DWDB**, and then expand **Tables**, and confirm [SalesLT].[ProductCategory] and [SalesLT].[ProductDescription] exist.
5. Close down **Microsoft SQL Server Management Studio**

**Result**: After you completed this exercise, you have created the Azure Data Factory components to move data from Azure SQL Database to Azure SQL Data Warehouse through the use of a wizard. You have confirmed the components used and that the data has loaded into the Data Warehouse.

**Exercise 3: Azure Data Factory and Databricks**

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Generate a Databricks Access Token.
2. Generate a Databricks Notebook
3. Create Linked Services
4. Create a Pipeline that uses Databricks Notebook Activity.
5. Trigger a Pipeline Run.

**Task 1: Generate a Databricks Access Token.**

1. In the Azure portal, click on **Resource groups** and then click on **awrgstudxx**, and then click on **awdbwsstudxx** where xx are the initials of your name.
2. Click on **Launch Workspace**
3. Click the user **profile icon** in the upper right corner of your Databricks workspace.
4. Click **User Settings**.
5. Go to the Access Tokens tab, and click the **Generate New Token** button.
6. Enter a description in the **comment** "For ADF Integration" and set the **lifetime** period of 10 days and click on **Generate**
7. Copy the generated token and store in Notepad, and then click on **Done**.

**Task 2: Generate a Databricks Notebook**

1. On the left of the screen, click on the **Workspace** icon, then click on the arrow next to the word Workspace, and click on **Create** and then click on **Folder**. Name the folder **adftutorial**, and click on **Create Folder**. The adftutorial folder appears in the Workspace.
2. Click on the drop down arrow next to adftutorial, and then click **Create**, and then click **Notebook**.
3. In the Create Notebook dialog box, type the name of **mynotebook**, and ensure that the language states **Python**, and then click on **Create**. The notebook with the title of mynotebook appears/
4. In the newly created notebook "mynotebook'" add the following code:

# Creating widgets for leveraging parameters, and printing the parameters

dbutils.widgets.text("input", "","")

dbutils.widgets.get("input")

y = getArgument("input")

print ("Param -\'input':")

print (y)

**Note** that the notebook path is **/adftutorial/mynotebook**

**Task 3: Create Linked Services**

1. In Microsoft Edge, click on the tab for the portal In the Azure portal, and return to Azure Data Factory.
2. In the **xx-data-factory** screen, click on **Author & Monitor**. Another tab opens up to author an Azure Data Factory solution.
3. On the left hand side of the screen, click on the **Author** icon. This opens up the Data Factory designer.
4. At the bottom of the screen, click on **Connections**, and then click on **+ New**.
5. In the **New Linked Service**, at the top of the screen, click on **Compute**, and then click on **Azure Databricks**, and then click on **Continue**.
6. In the **New Linked Service (Azure Databricks)** screen, fill in the following details and click on **Finish**
   * **Name**: xx\_dbls, where xx are your initials
   * **Databricks Workspace**: awdbwsstudxx, where xx are your initials
   * **Select cluster**: use existing
   * **Domain/ Region**: should be populated
   * **Access Token**: Copy the access token from Notepad and paste into this field
   * **Choose from existing cluster**: awdbclstudxx, where xx are your initials
   * Leave other options to their default settings

**Note**: When you click on finish, you are returned to the **Author & Monitor** screen where the xx\_dbls has been created, with the other linked services created in the previous exercise.

**Task 5: Create a pipeline that uses Databricks Notebook Activity.**

1. On the left hand side of the screen, under Factory Resources, click on the **+** icon, and then click on **Pipeline**. This opens up a tab with a Pipeline designer.
2. At the bottom of the pipeline designer, click on the parameters tab, and then click on **+ New**
3. Create a parameter with the Name of **name**, with a type of **string**
4. Under the **Activities** menu, expand out **Databricks**.
5. Click and drag **Notebook** onto the canvas.
6. In the properties for the **Notebook1** window at the bottom, complete the following steps:
   * Switch to the **Azure Databricks** tab.
   * Select **xx\_dbls** which you created in the previous procedure.
   * Switch to the **Settings** tab, and put **/adftutorial/mynotebook** in Notebook path.
   * Expand **Base Parameters**, and then click on **+ New**
   * Create a parameter with the Name of **input**, with a value of **@pipeline().parameters.name**
7. In the **Notebook1**, click on **Validate**, next to the Save as template button. As window appears on the right of the screen that states "Your Pipeline has been validated. No errors were found." Click on the >> to close the window.
8. Click on the **Publish All** to publish the linked service and pipeline.

**Note**: A message will appear to state that the deployment is successful.

**Task 6: Trigger a Pipeline Run**

1. In the **Notebook1**, click on **Add trigger**, and click on **Trigger Now** next to the Debug button.
2. The **Pipeline Run** dialog box asks for the name parameter. Use **/path/filename** as the parameter here. Click Finish. A red circle appear above the Notebook1 activity in the canvas.

**Task 7: Monitor the Pipeline**

1. On the left of the screen, click on the **Monitor** tab. Confirm that you see a pipeline run. It takes approximately 5-8 minutes to create a Databricks job cluster, where the notebook is executed.
2. Select **Refresh** periodically to check the status of the pipeline run.
3. To see activity runs associated with the pipeline run, select **View Activity Runs** in the **Actions** column.

**Task 8: Verify the output**

1. In Microsoft Edge, click on the tab **mynotebook - Databricks**
2. In the **Azure Databricks** workspace, click on **Clusters** and you can see the Job status as pending execution, running, or terminated.
3. Click on the cluster **awdbclstudxx**, and then click on the **Event Log** to view the activities.

**Note**: You should see an Event Type of **Starting** with the time you triggered the pipeline run.