Example - Use compute transformations within Azure Data Factory

Note:

You are not required to complete the processes, tasks, activities, or steps presented in this example. The various provided are for illustrative purposes only and it’s likely that if you try this out you will encounter issues in your

In some cases, the code-free transformation at scale may not meet your requirements. You can use Azure Data Factory to ingest raw data collected from different sources and work with a range of compute resources such as Azure Databricks, Azure HDInsight, or other compute resources to restructure it as per your requirements.

ADF and Azure Databricks

As an example, the integration of Azure Databricks with ADF allows you to add Databricks notebooks within an ADF pipeline to leverage the analytical and data transformation capabilities of Databricks. You can add a notebook within your data workflow to structure and transform raw data loaded into ADF from different sources. Once the data is transformed using Databricks, you can then load it to any data warehouse source.

Data ingestion and transformation using the collective capabilities of ADF and Azure Databricks essentially involves the following steps:

1. **Create Azure storage account** - The fist step is to create an Azure storage account to store your ingested and transformed data.
2. **Create an Azure Data Factory** - Once you have your storage account setup, you need to create your Azure Data Factory using Azure portal.
3. **Create data workflow pipeline** - After your storage and ADF is up and running, you start by creating a pipeline, where the first step is to copy data from your source using ADF's Copy activity. Copy Activity allows you to copy data from different on-premises and cloud sources.
4. **Add Databricks notebook to pipeline** - Once your data is copied to ADF, you add your Databricks notebook to the pipeline, after copy activity. This notebook may contain syntax and code to transform and clean raw data as required.
5. **Perform analysis on data** - Now that your data is cleaned up and structured into the required format, you can use Databricks notebooks to further train or analyze it to output required results.

You have learned what Azure Data Factory is and how its integration with Azure Databricks helps you to load and transform your data. Now let's create an end-to-end sample data workflow.

Integrating Azure Databricks notebooks with Azure Data Factory pipeline

There are a number of tasks that needs to be performed to integrate Azure Databricks notebooks with Azure Data Factory pipeline 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.

Note: The following steps assume there is already an Azure Databricks cluster already provisioned

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.

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 exercize.

Task 4: 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.
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 5: 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 6: 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 7: 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.