Orchestrating Big Data Solutions with Azure Data Factory

Lab 4 – Transforming Data with U-SQL

Note: If you prefer to transform data using Hive, an alternative version of this lab is provided in which you can use an HDInsight cluster to run a Hive script.

Overview

In this lab, you will use Azure Data Factory to implement a pipeline that uses U-SQL to transform web server log data before copying it to 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 labs

Important: If you have not completed Lab 1, or you have deleted the storage account, SQL database, and Azure Data Factory resources you created, complete Lab 1 now.

Exercise 1: Preparing for Data Transformation

In this exercise, you will prepare the environment for your data transformation pipeline, and explore the source data.

Upload the Source Data and U-SQL Script

The source data for this pipeline is a monthly log file from a web server. Your pipeline will use a Hive script to aggregate the log data to create daily summaries.

1. In the **iislogs** subfolder of the folder where you extracted the lab files for this course, open the **2016** subfolder and note that it contains six subfolders (**01** to **06**). These folders contain web server log files for the months of January to June in 2016.
2. In the **01** folder, open **log.txt** in a text editor, and examine the data it contains. After some initial header rows, the log files contain details of web server requests. After you have viewed the structure of the data, close the text editor without saving any changes.

3. In the **iislogs** folder, use a text editor to open the **SummarizeLogs.usql** script file, and note that it contains the following U-SQL script:

```
@log =
EXTRACT date string,
        time string,
        client_ip string,
        username string,
        server_ip string,
        port int,
        method string,
        stem string,
        query string,
        status string,
        server_bytes int,
        client_bytes int,
        time_taken int?,
        user_agent string,
        referrer string
FROM @log_file
USING Extractors.Text(' ', silent:true);

@summary =
SELECT date,
       COUNT(*) AS hits,
       SUM(server_bytes) AS bytes_in,
       SUM(client_bytes) AS bytes_out
FROM @log
GROUP BY date;

OUTPUT @summary
    TO @summary_file
    USING Outputters.Csv();
```

   This script:
   a. Extracts the log data from a space-delimited text file, which is passed to the script as a variable named **@log_file**.
   b. Creates a variable named **@summary** that is contains the results of a query that aggregates the log data by date.
   c. Stores the summarized results as a comma-separated values (CSV) file in a folder location, which is passed to the script as a variable named **@summary_file**.
4. Close the U-SQL script without saving any changes.
5. Start Azure Storage Explorer, and if you are not already signed in, sign into your Azure subscription.
6. Expand your storage account and the **Blob Containers** folder, and then double-click the **adf-data** blob container you created in a previous lab.
7. In the **Upload** drop-down list, click **Folder**. Then upload the **iislogs** folder as a block blob to a new folder named **iislogs** in the root of the container.

Create an Azure Data Lake Analytics Account

Your data processing requires an Azure Data Lake Analytics account to run the U-SQL script, and an Azure Data Lake Store in which to save the results. Additionally, your Azure Data Lake Analytics account requires access to the Azure Blob Store containing the log files to be processed.

1. In the Microsoft Azure portal, in the Hub Menu, click **New**. Then in the **Intelligence and analytics** menu, click **Data Lake Analytics**.
2. In the **New Data Lake Analytics Account** 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**: *Create a new resource group with a unique name*
    - **Location:** *Select any available region*
    - **Data Lake Store:** *Create a new Data Lake Store with a unique name (and make a note of it!)*
    - **Pin to dashboard**: *Not selected*
3. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the resources to be deployed (this can take a few minutes.)
4. After the Azure Data Lake Analytics account has been provisioned, browse to its blade in the Azure portal, and under **Settings**, click **Data Sources**.
5. Click **Add Data Source**. Then in the **Add Data Source** blade, in the **Storage Type** list, select **Azure Storage**, and then select your Azure storage account. This adds your Azure storage account as a data source to which the Azure Data Lake Analytics account has access, in addition to its default Azure Data Lake Store.

Prepare the Database

The data your pipeline will copy contains daily summaries of the web server log file data. You will copy this data to a database table named **dbo.usql_logs**.

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 dbo.usql_logs (log_date varchar(12), requests int,
    bytes_in float, bytes_out float);
    ```

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

Exercise 2: Creating Linked Services

The pipeline for your data transformation will use the existing linked services for Azure Blob storage and Azure SQL database that you created in a previous lab. It will also use a new linked service for an on-demand Azure HDInsight cluster, on which the Hive script to transform the data will be run.

Verify Existing Linked Services

You require linked services for the blob store account where the log data is stored, and the Azure SQL Database containing the table you want to load. You should have created these linked services in the previous lab.

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 a 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 Azure Data Lake Linked Services

In addition to the Azure blob store and Azure SQL Database, your pipeline will need to use the Azure Data Lake Analytics service to run the U-SQL script, and the Azure Data Lake Store in which the results of the U-SQL processing will be stored.

1. In the **Linked services / sql-database** blade (containing the JSON for your **sql-database** linked service), click **Clone** to create a copy of the JSON document in the **Drafts** section. Then click **Discard** to close the copied JSON and create a blank document.

2. In the new blank document, enter the following code, which you can copy and paste from **adl-analytics.json** in the folder where you extracted the lab files). Replace ***<Azure Data Lake Analytics account>*** with the name of your Azure Data Lake Analytics account:

```
{
    "name": "adl-analytics",
    "properties": {
        "type": "AzureDataLakeAnalytics",
        "typeProperties": {
          "authorization": "<Authorization code …>",
          "accountName": "<Azure Data Lake Analytics account>",
          "sessionId": "<OAuth session id from the OAuth …>"
      }
    }
}
```

   This JSON defines an Azure Data Lake Analytics account. The linked service must be authorized to access the account.

3. Click **Authorize**, and when prompted enter your Microsoft account credentials to sign into your Azure subscription – this will verify your identity and generate the authorization code and session ID in the JSON document.
4. Click **Deploy** to deploy the linked service definition to your Azure Data Factory.
5. In the new blank document, enter the following code, which you can copy and paste from **adl-store.json** in the folder where you extracted the lab files). Replace ***<store_name>*** with the name of your Azure Data Lake Store:

```
{
    "name": "adl-store",
    "properties": {
        "type": "AzureDataLakeStore",
        "description": "",
        "typeProperties": {
          "authorization": "<Click 'Authorize' …>",
          "dataLakeStoreUri":
"https://<store_name>.azuredatalakestore.net/webhdfs/v1",
          "sessionId": "<OAuth session id from the OAuth …>"
      }
    }
}
```

   This JSON defines an Azure Data Lake Analytics Store. The linked service must be authorized to access the store.

6. Click **Authorize**, and when prompted enter your Microsoft account credentials to sign into your Azure subscription – this will verify your identity and generate the authorization code and session ID in the JSON document.
7. Click **Deploy** to deploy the linked service definition to your Azure Data Factory.

Exercise 3: Creating Datasets

The pipeline for your data transformation requires three datasets; one to define the source data, one to define the results of the aggregation transformation produced by the U-SQL script, and one to define the destination table in Azure SQL Database.

Create the Source Dataset

The source dataset defines the data in the web server log files in Azure blob storage.

1. In the new blank document, enter the following code, which you can copy and paste from **usql-iis-log.json** in the folder where you extracted the lab files:

```
{
  "name": "usql-iislog-txt",
  "properties": {
    "structure": [
      {
        "name": "log_date",
        "type": "String"
      },
      {
        "name": "log_time",
        "type": "String"
      },
      {
        "name": "c_ip",
        "type": "String"
      },
      {
        "name": "cs_username",
        "type": "String"
      },
      {
        "name": "s_ip",
        "type": "String"
      },
      {
        "name": "s_port",
        "type": "String"
      },
      {
        "name": "cs_method",
        "type": "String"
      },
      {
        "name": "cs_uri_stem",
        "type": "String"
      },
      {
        "name": "cs_uri_query",
        "type": "String"
      },
      {
        "name": "sc_status",
        "type": "String"
```

```
    },
    {
      "name": "sc_bytes",
      "type": "Int32"
    },
    {
      "name": "cs_bytes",
      "type": "Int32"
    },
    {
      "name": "time_taken",
      "type": "Int32"
    },
    {
      "name": "cs_user_agent",
      "type": "String"
    },
    {
      "name": "cs_referrer",
      "type": "String"
    }
  ],
  "type": "AzureBlob",
  "linkedServiceName": "blob-store",
  "typeProperties": {
    "folderPath": "adf-data/iislogs/{Year}/{Month}/",
    "format": {
      "type": "TextFormat",
      "columnDelimiter": " "
    },
    "partitionedBy": [
      {
        "name": "Year",
        "value": {
          "type": "DateTime",
          "date": "SliceStart",
          "format": "yyyy"
        }
      },
      {
        "name": "Month",
        "value": {
          "type": "DateTime",
          "date": "SliceStart",
          "format": "MM"
        }
      }
    ]
  },
  "availability": {
    "frequency": "Month",
    "interval": 1
  },
  "external": true,
  "policy": {
```

```
      "validation": {
        "minimumSizeMB": 0.01
      }
    }
  }
}
```

This JSON defines a schema for the space-delimited log files in the **iislogs/*Year*/*Month*** folder hierarchy in the **adf-data** container of the Azure storage account represented by your **blob-store** linked service. New data will be available every month.

2. Click **Deploy** to deploy the dataset definition to your Azure Data Factory.

Create a Dataset for the Summarized Data File

The U-SQL job transforms the source data by aggregating it, and stores the results in a text file in Azure Data Lake Store.

1. In the new blank document, enter the following code, which you can copy and paste from **usql-summary.json** in the folder where you extracted the lab files:

```
{
  "name": "usql-summary",
  "properties": {
    "structure": [
      {
        "name": "log_date",
        "type": "String"
      },
      {
        "name": "requests",
        "type": "Int64"
      },
      {
        "name": "bytes_in",
        "type": "Decimal"
      },
      {
        "name": "bytes_out",
        "type": "Decimal"
      }
    ],
    "published": false,
    "type": "AzureDataLakeStore",
    "linkedServiceName": "adl-store",
    "typeProperties": {
      "fileName": "summary.txt",
      "folderPath": "iislogs/summary/{Year}/{Month}",
      "format": {
        "type": "TextFormat",
        "columnDelimiter": ","
      },
      "partitionedBy": [
        {
          "name": "Year",
          "value": {
```

```
            "type": "DateTime",
            "date": "SliceStart",
            "format": "yyyy"
          }
        },
        {
          "name": "Month",
          "value": {
            "type": "DateTime",
            "date": "SliceStart",
            "format": "MM"
          }
        }
      ]
    },
    "availability": {
      "frequency": "Month",
      "interval": 1
    }
  }
}
```

This JSON defines a schema for the files generated by the U-SQL script. These files are saved in a folder hierarchy that includes subfolders for each year and month.

2. Click **Deploy** to deploy the dataset definition to your Azure Data Factory.

Create the Database Table Dataset

The summarized data is copied to the **dbo.usql_logs** table in Azure SQL Database.

1. In the new blank document, enter the following code, which you can copy and paste from **dbo-usql_logs.json** in the folder where you extracted the lab files:

```
{
    "name": "dbo-usql_logs",
    "properties": {
        "type": "AzureSqlTable",
        "linkedServiceName": "sql-database",
        "structure": [
            {
                "name": "log_date",
                "type": "String"
            },
            {
                "name": "requests",
                "type": "Int32"
            },
            {
                "name": "bytes_in",
                "type": "Decimal"
            },
            {
                "name": "bytes_out",
                "type": "Decimal"
            }
```

```
        ],
        "typeProperties": {
            "tableName": "dbo.usql_logs"
        },
        "availability": {
            "frequency": "Month",
            "interval": 1
        }
    }
}
```

This JSON defines a schema for the **dbo.usql_logs** table you created previously in the database defined by the **sql-database** linked service.

2. Click **Deploy** to deploy the dataset definition to your Azure Data Factory.
3. In the pane on the left, expand the **Datasets** folder and verify that the **usql-iislog-txt**, **usql-summary**, and **dbo-usql_logs** datasets are listed.

Exercise 4: Creating and Running the Pipeline

Now that you have defined the linked services and datasets for your data flow, you can create a pipeline to encapsulate it.

Create the Pipeline

In this lab, your pipeline will consist of a **DataLakeAnalyticsU-SQL** action to summarize the web server log data, followed by a **Copy** action to copy the summarized results to Azure SQL Database.

1. In the new blank document, enter the following code, which you can copy and paste from **summarize-logs-usql.json** in the folder where you extracted the lab files:

   Important: Replace ***<storage_acct>*** with the name of your Azure storage account:

```
{
  "name": "Summarize Logs - U-SQL",
  "properties": {
    "activities": [
      {
        "type": "DataLakeAnalyticsU-SQL",
        "typeProperties": {
          "scriptPath": "adf-data/iislogs/SummarizeLogs.usql",
          "scriptLinkedService": "blob-store",
          "degreeOfParallelism": 2,
          "parameters": {
            "log_file": "$$Text.Format('wasb://adf-
data@<storage_acct>.blob.core.windows.net/iislogs/{0:yyyy}/{1:MM}/log.t
xt', SliceStart, SliceStart)",
            "summary_file":
"$$Text.Format('iislogs/summary/{0:yyyy}/{1:MM}/summary.txt',
SliceStart, SliceStart)"
          }
        },
        "inputs": [
          {
            "name": "usql-iislog-txt"
          }
```

```
        ],
        "outputs": [
          {
            "name": "usql-summary"
          }
        ],
        "policy": {
          "timeout": "01:00:00",
          "concurrency": 2,
          "executionPriorityOrder": "OldestFirst",
          "retry": 2
        },
        "scheduler": {
          "frequency": "Month",
          "interval": 1
        },
        "name": "U-SQL Script to Summarize Logs",
        "linkedServiceName": "adl-analytics"
      },
      {
        "type": "Copy",
        "typeProperties": {
          "source": {
            "type": "AzureDataLakeStoreSource",
            "recursive": false
          },
          "sink": {
            "type": "SqlSink",
            "writeBatchSize": 0,
            "writeBatchTimeout": "00:00:00"
          },
          "translator": {
            "type": "TabularTranslator",
            "columnMappings":
"log_date:log_date,requests:requests,bytes_in:bytes_in,bytes_out:bytes_
out"
          }
        },
        "inputs": [
          {
            "name": "usql-summary"
          }
        ],
        "outputs": [
          {
            "name": "dbo-usql_logs"
          }
        ],
        "policy": {
          "timeout": "01:00:00",
          "concurrency": 2,
          "executionPriorityOrder": "OldestFirst",
          "retry": 2
        },
        "scheduler": {
```

```
            "frequency": "Month",
            "interval": 1
          },
          "name": "Copy summarized data to SQL"
        }
      ],
      "start": "2016-01-01T00:00:00Z",
      "end": "2016-06-01T23:59:59Z",
      "pipelineMode": "Scheduled"
    }
}
```

This JSON defines a pipeline that includes a **DataLakeAnalyticsU-SQL** action to run the U-SQL script that transforms the **usql-iislog-txt** dataset in your Azure blob storage account to the **usql-summary** dataset in the Azure Data Lake store, and a **Copy** action to copy the **usql-summary** dataset to the **dbo.usql_logs** table in Azure SQL Database every month. Because the input dataset to the second activity is the output dataset from the first activity, the pipeline will start the second activity only after the first activity has completed successfully.

The **DataLakeAnalyticsU-SQL** action defines two parameters that are passed to the Hive script:

- **log_file**: The path to the log file blob in your Azure storage account.
- **summary_file**: The path where the summarized results are to be stored in Azure Data Lake.

Both parameters include year and month variables that will reflect the current time slice when the activity is run.

2. 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, wait a few minutes for the **Pipelines** tile to indicate the addition of your pipeline (it may already indicate the pipelines you created in the previous lab).
2. Click the **Monitor and Manage** tile. This opens a new tab in your browser.
3. View the pipeline diagram, which should include the **Summarize and Copy Logs** pipeline that transfers data from the **usql-iislog-txt** Azure Blob Storage dataset to the **dbo-usql_logs** Azure SQL database dataset.
4. Right-click the **Summarize and Copy Logs** pipeline and click **Open pipeline**. The diagram should now show the **Hive script to summarize logs** and **Copy summarized logs to SQL** actions in the pipeline and the intermediary **usql-summary** Azure Data Lake Store dataset.
5. Above the diagram, click the start time and change it to 0:00 on January 1st 2016. Then click the end time and change it to 0:00 on July 1st 2016. Click **Apply** for the filter to take effect.
6. Wait for 30 minutes or so, as the HDInsight cluster is provisioned and the pipeline activities are run. You can refresh the list of activity windows periodically to see the status of each activity window change from **Waiting**, to **In progress**, and then to **Ready**. You can select each activity window in the list to see more details about its status in the pane on the right.

View the Output Generated by the Pipeline

When the first U-SQL actions has finished, you can view the summarized log files that it generates; and after the first of the copy actions has completed, you can verify that data is being copied to your SQL database.

1. In the Azure portal, browse to your Azure Data Lake Store and click **Data Explorer**.
2. In the **iislog** folder, verify that a subfolder named **summary** contains a subfolder named **2016**, and that this folder contains folders named **01** to **06** representing each month (the folders for the most recent month will be created first). Each monthly folder contains a text file named **summary.txt** containing the daily log summary data for that month.
3. In your SQL Server client tool, enter the following query:

   ```
   SELECT * FROM dbo.usql_logs ORDER BY log_date;
   ```

4. Verify that the table now contains aggregated log data. This was copied to the table from the summary log data files generated by the Hive script.
5. Keep viewing the summary log data files and querying the **dbo.usql_logs** table as the pipeline actions are processed. Eventually, data for all six months should have been summarized and copied to the database.
6. Close Azure Storage Explorer and the SQL Server client tool:

   To exit *mssql*, enter the following command:

   ```
   .quit
   ```