Big Data Orientation

Lab – Working with Data in Microsoft Azure

Overview

Before you start to design and build big data processing solutions, you should familiarize yourself with the Microsoft Azure portal and some of the key Azure resources that you will need to provision and use to store and process data.

In this lab, you will explore some commonly used Azure data storage services, and use them to store and manipulate data.

Before You Start

To complete this lab, you will need the following:

- A web browser
- A Windows, Linux, or Mac OS X computer

Before you can perform the lab exercises, you will also need:

- A Microsoft account
- A Microsoft Azure subscription
- The lab files for this course

Sign up for a Microsoft Account

You need a Microsoft account, such as an *outlook.com* or *live.com* address, in order to sign up for a Microsoft Azure subscription. If you already have a Microsoft account that has not already been used to sign up for a free Azure trial subscription, you're ready to get started. If not, don't worry, just create a new Microsoft account at https://signup.live.com.

Sign up for a Microsoft Azure Subscription

If you already have a Microsoft Azure subscription, you can skip this section, download the lab files, and get started.

If you do not have an Azure subscription, you can sign up for the Visual Studio Dev Essentials program at https://aka.ms/edx-dat229x-vsde and then activate a free 12-month Azure subscription. This will give you $25 of Azure credit per month for a year.

Alternatively, you can create a free 30-day trial subscription that includes approximately $200 credit in your local currency at https://aka.ms/edx-dat229x-az. The credit in this trial subscription expires after 30 days (after which time you can purchase a full Azure subscription, or sign up for the Visual Studio Dev Essentials program as described above).

Note: Some Azure resources consume credit even when not in use, so be careful to delete resources if you don't intend to use them for a while; otherwise you will run out of credit before the month ends.

Whichever option you use to sign up for a free Azure subscription, you will need to provide a valid credit card number for verification, but you will not be charged for Azure services – for more information, see the frequently asked questions in the Azure sign-up page.

Download the Lab Files

The files you'll need for this lab are provided in a zip file at https://aka.ms/edx-dat229x-labfiles. Download and extract the lab files to a folder on your local computer.

Exercise 1: Working with Azure Storage

Azure Storage is a general purpose storage service for storing data in Azure. It is widely used by applications and can provide distributed storage services for Big Data technologies. In this exercise, you will create an Azure Storage account and use it to store files.

Provision an Azure Storage Account and a Blob Container

You will use an Azure storage account to store data files. The files will be stored as binary large objects (BLOBs) in a blob store container that is hosted in your storage account.

1. In a web browser, navigate to http://portal.azure.com, and if prompted, sign in using the Microsoft account that is associated with your Azure subscription.
2. In the Microsoft Azure portal, in the Hub Menu (on the left edge of the page), click **New**. Then in the **Storage** menu, click **Storage account**.
3. In the **Create storage account** blade, enter the following settings, and then click **Create**:
    - **Name**: *Enter a unique name for your storage account (and make a note of it!)*
    - **Deployment model:** Resource manager
    - **Account kind**: General purpose
    - **Performance:** Standard
    - **Replication:** *Select **Locally redundant storage (LRS)***
    - **Storage service encryption**: Disabled
    - **Subscription**: *Your subscription*
    - **Resource group**: *Select **Create New** and enter a name for a new resource group (and make a note of it!)*
    - **Location:** *Select any available region*
    - **Pin to dashboard:** *Unselected*
4. At the top of the page, click **Notifications** and verify that deployment has started. Wait until your storage account has been created. This should take a few minutes.
5. In the Hub menu, click **All resources**, and then click your storage account.
6. In the blade for your storage account, click the **Blobs** tile, and note that you don't have any containers yet.
7. In the blob service blade, click **+Container**, and create a new container with the following properties:
    - **Name**: bigdata
    - **Access type**: Private
8. After the **bigdata** blob container has been created, click it and verify that it contains no blobs.
9. In the bigdata blade, click **Properties** and view the **URL** for the blob container, which should be in the form **https://<*your_account_name*>.blob.core.windows.net/bigdata**. This is the URL that client applications can use to access your blob container using HTTP protocol.

> **Note**: Azure blob storage also supports the WASB protocol, which is specific to Azure storage. Some big data processing technologies use this protocol to work with Azure storage.

10. Return to the blade for your storage account, and under **Settings**, click **Access keys**. Then on the **Access Keys** page, note that two keys have been generated. These are used to authenticate client applications that connect to your storage account.

Use the Azure Portal to Upload a File to Azure Storage

The Azure portal includes a rudimentary graphical interface that you can use to work with your Azure storage account. You can use this to transfer files between your local computer and your blob containers, and to browse the data in your storage account.

1. In the blade for your storage account, view the **Overview** page, and then click the **Blobs** tile.
2. Click the **bigdata** container that you created previously, and then click **Upload**.
3. In the **Upload blob** blade, browse to the folder where you extracted the lab files for this course, and select **products.txt**. Then verify the following settings and click **Upload**:
   - **Files**: “products.txt”
   - **Blob type**: Block blob
   - **Block size**: 100 MB

> **Note**: Azure storage supports three blob types (*block*, *page*, and *append*). Block blobs are formed of one or more blocks of data based on the specified block size, and provide an efficient format for uploading and transferring blobs. For more details about blob types in Azure storage, see https://docs.microsoft.com/en-us/rest/api/storageservices/fileservices/Understanding-Block-Blobs--Append-Blobs--and-Page-Blobs.

4. After the blob has been uploaded, note that it is listed in the **bigdata** container blade. Then click the **Products.txt** blob and in the **Blob** properties blade, note its **URL**, which should be similar to **https://<*your_account_name*>.blob.core.windows.net/bigdata/products.txt**.

Use Azure Storage Explorer to Upload files to Azure Storage

The blob container interface in the Azure portal enables you to upload, browse, and download blobs; but it lacks many of the features expected in a modern file management tool. There are various graphical Azure storage management tools available, including support for exploring your Azure storage in Microsoft Visual Studio. However, if you do not need the full Visual Studio environment, you can install Azure Storage Explorer, which is available for Windows, Mac OSX, and Linux.

1. Open a new browser tab and browse to http://storageexplorer.com.
2. Download and install the latest version of Azure Storage Explorer for your operating system (Windows, Mac OSX, or Linux).
3. When the application is installed, launch it. Then add your Azure account, signing in with your Azure credentials when prompted, and configure Storage Explorer to show resources from the Azure subscription in which you created your storage account.
4. After your subscription has been added to the **Explorer** pane expand your storage account, expand **Blob Containers**, and select the **bigdata** container. Note that the **products.txt** file you uploaded previously is listed.
5. In the **Upload** drop-down menu, note that you can choose to upload individual files or folders. Then select **Upload Folder** and browse to the folder where you extracted the

lab files for this course and select the **data** folder, and upload it as a block blob to a folder named **data** in the blob container.

6. After the upload operation is complete, double-click the **data** folder in your blob container to open it, and verify that it contains files named **customers.txt** and **reviews.txt**.
7. Click the ↑ button to navigate back up to the root of the **bigdata** container, and select the **products.txt** file. Then click **Copy**.
8. Open the **data** folder, and then click **Paste** to copy the **product.txt** file to this folder.
9. Navigate back up to the root of the **bigdata** container, and select the **products.txt** file. Then click **Delete**, and when prompted to confirm the deletion, click **Yes**.
10. Verify that the **bigdata** container now contains only a folder named **data**, which in turn contains files named **customers.txt**, **products.txt**, and **reviews.txt**.

Exercise 2: Working with Azure Data Lake Store

Azure Data Lake Store is a storage service in Azure that is optimized for big data workloads. It supports unlimited numbers of files of unlimited size and can be used to organize and secure files in folder hierarchies. In this exercise, you will provision Azure Data Lake Store and upload some files to it.

Note: For a detailed comparison of Azure Storage and Azure Data Lake Store, see https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-comparison-with-blob-storage.

Provision Azure Data Lake Store

To get started, you must provision Azure Data Lake Store.

1. In the Microsoft Azure portal, in the Hub Menu (on the left edge of the page), click **New**. Then in the **Storage** menu, click **Data Lake Store**.
2. In the **New Data Lake Store** blade, enter the following settings, and then click **Create**:
   - **Name**: *Enter a unique name for your storage account (and make a note of it!)*
   - **Resource group**: *Select **Use existing** and select the resource group you created previously.*
   - **Location:** *Select any available region*
   - **Pricing:** *Pay-as-you-go*
   - **Encryption Settings:** Enabled
   - **Pin to dashboard:** *Unselected*
3. At the top of the page, click **Notifications** and verify that deployment has started. Wait until your storage account has been created. This should take a few minutes.

Upload Files to Azure Data Lake Store

Now that you have provisioned Azure Data Lake Store, you can use the Data Explorer in the Azure Portal to upload files.

1. In the Hub menu, click **All resources**, and then click your Data Lake Store.
2. In the blade for your Data Lake Store, click **Data Explorer**.
3. In the Data Explorer blade, click **New Folder**. Then create a folder named **data**.
4. Click the **data** folder to open it.
5. Click **Folder Properties**; and in the **Properties** blade, view the PATH property. This is the folder URL, and should be similar to **adl://<*your_account_name*>.azuredatalakestore.net/data/**
6. In the blade for the **data** folder, click **Upload**.
7. In the **Upload files** blade, browse to the folder where you extracted the lab files for this course and view the contents of the **data** folder. Then select all the files it contains (hold the CTRL key

to select multiple files) and click **Add selected files** to upload them to the **data** folder in your Azure Data Lake Store.

8. After the files have been uploaded, close the Upload files blade and verify that the data folder in your Azure Data Lake Store now contains **customers.txt** and **reviews.txt**.
9. Click **reviews.txt** and view the preview of the data. Note that the tab-delimiter has been detected and the data is shown as a table with multiple columns.

Exercise 3: Working with Azure SQL Database

Azure SQL Database is a cloud service based on the Microsoft SQL Server relational database management system (RDBMS). Application developers can use Azure SQL Database as a relational store for application data, which can be used in big data solutions. In addition to Azure SQL Database, Azure includes a data warehouse service named Azure SQL Data Warehouse, which shares the same core database engine as Azure SQL Database but is optimized for large data workloads, and often provides an analytical data store into which big data processing solutions load the processed data for analysis and reporting.

In this exercise, you will provision and work with Azure SQL Database. The tasks you will perform in this exercise can also be performed with Azure SQL Data Warehouse.

Provision Azure SQL Database

To get started, you must provision Azure SQL Database.

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**: AdventureWorksLT
   - **Subscription**: *Select your Azure subscription*
   - **Resource Group**: *Select the resource group you created previously*
   - **Select source**: Sample (AdventureWorksLT)
   - **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!)*
     - **Location:** *Select your HDInsight cluster location*
     - **Allow azure services to access server**: Selected
   - **Want to use SQL elastic pool?** Not now.
   - **Pricing tier:** 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 AdventureWorksLT 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).

Query a Table

A relational database contains tables, each of which contains data. Tables are organized into namespaces called *schemas* – in the case of the **AdventureWorksLT** sample database, most of the tables are defined within a schema named **SalesLT**.

You can query tables using Transact-SQL to retrieve the data they contain.

1. Click **All Resources**, and then click the **AdventureWorksLT** database.
2. On the **AdventureWorksLT** blade, click **Tools**. Then on the **Tools** blade, click **</> Query editor**. 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. In the query editor, enter the following Transact-SQL query to retrieve the contents of the **SalesLT.Product** table in the **AdventureWorksLT** database:

   ```
   SELECT * FROM SalesLT.Product;
   ```

6. Click **Run**, and review the results.

Create a Table

The sample database contains many tables, and you can add your own by using the Transact-SQL CREATE TABLE statement.

1. In the Query pane, replace the existing SELECT statement with the following code:

   ```
   CREATE TABLE SalesLT.ProductReview
   ( ProductReviewID INTEGER PRIMARY KEY,
     ProductID INTEGER REFERENCES SalesLT.Product(ProductID),
     ReviewerName NVARCHAR(25),
     ReviewDate DATETIME,
     EmailAddress NVARCHAR(50),
     Rating INTEGER,
     Comments NTEXT );
   ```
2. Click **Run**, and verify that the query succeeds.
3. Replace the CREATE TABLE statement with the following SELECT statement:

   ```
   SELECT * FROM SalesLT.ProductReview;
   ```

4. Click **Run**, and verify that the query succeeds but returns 0 rows.
5. Close the query editor without saving any changes.

Load Data from a File into the Table

A common task in a big data solution is to transfer data from one store to another. In this case, you will use the Copy Data wizard in the Azure Data Factory service to load the product review data from a text file in Azure Storage into the table you created in Azure SQL Database.

1. In the Microsoft Azure portal, in the menu, click **New**. Then in the **Data + 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. View **Notifications** to verify that deployment has started. Then wait for the data factory to be deployed (this can take a few minutes.)
4. Click **All Resources**, and then click your data factory, and click the **Copy data** tile. This opens a new tab in your browser.
5. On the **Properties** page of the Copy Data wizard, enter the following details and then click **Next**:
   - **Task name**: Load Reviews
   - **Task description**: Load review data into Azure SQL Database
   - **Task cadence (or) Task schedule**: Run once now
   - **Expiration time**: 3:00:00:00
6. On the **Source data store** page, on the **Connect to a Data Store** tab, select **Azure Blob Storage**. Then click **Next**.
7. 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*
8. On the **Choose the input file or folder** page, double-click the **bigdata** blob container you created previously and the **data** folder, and select the **reviews.txt** file. Then click **Choose**, and click **Next**.
9. 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**: Tab (\t)
   - **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
10. On the **Destination data store** page, on the **Connect to a Data Store** tab, select **Azure SQL Database**. Then click **Next**.
11. 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:** AdventureWorksLT
    - **User name**: *The server admin login name you specified when creating the database*
    - **Password**: *The password for your Azure SQL server admin login*
12. On the **Table mapping** page, in the **Destination** list, select **[SalesLT].[ProductReview]** and click **Next**.
13. On the **Schema mapping** page, ensure that the following settings are selected, and click **Next**:

Blob path: adf-data/data/	**[SalesLT].[ProductReview]**	**Include this column**

ProductReviewID (Int64)	ProductReviewID (Int32)	✓
ProductID (Int64)	ProductID (Int32)	✓
ReviewerName (String)	ReviewerName (String)	✓
ReviewDate (DateTime)	ReviewDate (DateTime)	✓
EmailAddress (String)	EmailAddress (String)	✓
Rating (Int64)	Rating (Int32)	✓
Comments (String)	Comments (String)	✓
Repeatability settings:		
Method: None		

14. On the **Performance settings** page, expand **Advanced settings** to review the default values. Then click **Next**.
15. On the **Summary** page, click **Finish**.
16. 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. Azure Data Factory may take several minutes to schedule the pipeline to run, so wait for around 5 minutes.

1. Click **All Resources**, and then click the **AdventureWorksLT** database.
2. On the **AdventureWorksLT** blade, click **Tools**. Then on the **Tools** blade, click **</> Query editor**. 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. In the query editor, enter the following Transact-SQL query to retrieve the contents of the **SalesLT.ProductReview** table in the **AdventureWorksLT** database:

   ```
   SELECT * FROM SalesLT.ProductReview;
   ```

5. Click **Run**, and review the results.

Cleaning Up

Now that you have finished this lab, you can delete the Azure resources you have been using.

Delete the Azure Resource Group

If you no longer need the Azure resources used in this lab, you should delete them to avoid incurring unnecessary costs.

1. In the Microsoft Azure portal, click **Resource Groups**.
2. On the **Resource groups** blade, click the resource group that contains your resources, and then on the **Resource group** blade, click **Delete**. On the confirmation blade, type the name of your resource group, and click **Delete**.
3. Wait for your resource group to be deleted, and then click **All Resources**, and verify that the resources have all been removed.
4. Close the browser.