Contents

1 DP-200-Implementing-an-Azure-Data-Solution

During this course, the first and the last lab of the course are group exercises that involve discussion to help provide context for the labs that the students will take. The last lab provides the opportunity for the students to reflect on what they have achieved and what they have overcome to achieve the delivery of requirements from the case study in the labs. The rest of the labs are hands on implementing Azure data platform capabilities to meet AdventureWorks business requirements.

The following is a summary of the lab objectives for each module:

1.1 Lab 1 - Azure for the Data Engineer

The students will take the information gained in the lessons and from the case study to scope out the deliverables for a digital transformation project within AdventureWorks. They will first identify how the evolving use of data has presented new opportunities for the organization. The students will also explore which Azure Data Platform services can be used to address the business needs and define the tasks that will be performed by the data engineer. Finally, students will finalize the data engineering deliverables for AdventureWorks.

1.2 Lab 2 - Working with Data Storage

In this lab, the students will be able to determine the appropriate storage type to implement against a given set of business and technical requirements. They will be able to create Azure storage accounts and Data Lake Storage account and explain the difference between Data Lake Storage version 1 and version 2. They will also be able to demonstrate how to perform data loads into the data storage of choice.

1.3 Lab 3 - Enabling Team Based Data Science with Azure Databricks

By the end of this lab the student will be able to explain why Azure Databricks can be used to help in Data Science projects. The students will provision and Azure Databricks instance and will then create a workspace that will be used to perform a simple data preparation task from a Data Lake Store Gen II store. Finally, the student will perform a walk-through of performing transformations using Azure Databricks.

1.4 Lab 4 - Building Globally Distributed Databases with Cosmos DB

The students will be able to describe and demonstrate the capabilities that Azure Cosmos DB can bring to an organization. They will be able to create a Cosmos DB instance and show how to upload and query data through a portal and through a .Net application. They will then be able to demonstrate how to enable global scale of the Cosmos DB database.

1.5 Lab 5 - Working with Relational Data Stores in the Cloud

The students will be able to provision an Azure SQL Database and Azure Synapse Analytics to be able to issue queries against one of the instances that are created. They will be also be able to integrate Azure Synapse Analytics with a number of other Data platform technologies and use PolyBase to load data from one data source into a data warehouse.

1.6 Lab 6 - Performing Real-Time Analytics with Stream Analytics

The students will be able to describe what data streams are and how event processing works and choose an appropriate data stream ingestion technology for the AdventureWorks case study. They will provision the chosen ingestion technology and integrate this with Stream Analytics to create a solution that works with streaming data.

1.7 Lab 7 - Orchestrating Data Movement with Azure Data Factory

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.

1.8 Lab 8 - Securing Azure Data Platforms

The students will be able to describe and document the different approaches to security that can be taken to provide defence in depth. This will involve the student documenting the security that has been set up so far in the course. It will also enable the students to identify any gaps in security that may exists for AdventureWorks.

1.9 Lab 9 - Monitoring and Troubleshooting Data Storage and Processing

The students will be able to define a broad monitoring solution that can help them monitor issues that can occur in their data estate. The student will then experience common data storage issues and data processing issue that can occur in cloud data solution. Finally they will implement a disaster recovery approach for a Data Platform technology.

2 Case Study – AdventureWorks Cycles

AdventureWorks sells bicycles and bicycle parts directly to customers and distributors. The company currently has a single office in the Netherlands, and have been selling bicycles in the United States, Germany and Spain through a chain of distributors and through online sales on its website. The fulfilment of delivery is done by local distribution centers.

The company is planning to expand by establishing new offices because the sales growth in these countries has been increasing over the last 3 years. The locations are:

- Tokyo, Japan
- Seattle, USA
- Chicago, USA
- Berlin, Germany
- Barcelona, Spain
- Paris, France

In a highly competitive market, in which AdventureWorks has been in business for the last 15 years, it wants to become the most innovative bicycle company, providing both current and future bicycle owners with best in class technology and service that provides unique experiences.

The Research and Development department of AdventureWorks has successfully conceived the next wave of innovative products, and they are relying on Data Engineers, AI Engineers and Data Scientists to assist with both the design and implementation of the solution.

Given the increased level of sales and expansion at global scale, the existing data infrastructure won't meet the overall business requirements or the future growth that AdventureWorks aspires to. The Chief Information and Technology Officers have expressed the desire to abandon existing on-premises systems and move to the cloud to meet the growth expected. This is supported by the CFO as there has been a request for replacement hardware as the existing infrastructure comes to its end of life. The CFO is aware that the cloud could offer alternatives that are more cost efficient.

As a Senior Data Engineer, you will assist AdventureWorks in the solution design and implementation to meet the business, functional and technical requirements that the company has set forth to be successful for growth, expansion, and innovation strategies. You will execute this in a way that minimizes operational costs and can be monitored for effectiveness.

In a discovery workshop you ascertained the following information:

2.1 AdventureWorks Website

The web developers at AdventureWorks are transferring the existing website from an on-premises instance of IIS, to an Azure Web App. They have requested that a data store is made available that will hold the images of the products that are sold on the website.

2.2 Current Sales / Ordering system

The current software on which bicycle purchases are tracked, is a web-based application which directly stores order information into an on-premises SQL Server database named AdventureWorks2012. The current application is deployed with high-availability provided by SQL Server 2012 Always-on Availability groups. Due to global expansion and data governance requirements, AdventureWorks will transition this system to better serve their customers and will be looking for global availability of its application and data sales and ordering purposes, particularly during the months of November and December when demand for bikes grow ahead of the holiday period.

2.3 Data Analysis

The business reporting is currently being provided by a single on-premises database that is configured as a data warehouse, it holds a database named AdventureWorksDW which is used to provide historical reporting and descriptive analytics. In recent times, that server has been struggling to process the reporting data in a timely manner, as a result the organization has evaluated the data warehouse capabilities of Azure Synapse Analytics and want to migrate their on-premises data to this platform. Your team should ensure that access to the data is restricted.

In addition, AdventureWorks would like to take their data analytics further and start to utilize predictive analytics capabilities. This is currently not an activity that is undertaken. The organization understands that a recommendation or a text analytics engine could be built and would like you to direct them on what would be the best technology and approach to take in implementing such a solution that is also resilient and performant.

You are also assessing the tooling that can help with the extraction, load and transforming of data into the data warehouse, and have asked a Data Engineer within your team to show a proof of concept of Azure Data Factory to explore the transformation capabilities of the product

2.4 Customer Service / Presales

Customer service and pre-sales departments are currently experiencing scale issues due to the high call volumes. The organization wants to support the customer services staff in handling the call volumes through the implementation of chat bots in which future bicycle owners can:

- Find which bicycle is best for them:
  - Through a set of questions with the chat bot, custom recommendations are given to potential bike owners, who then can take the recommendation and place an order, or can be redirect to a sales specialist to help them with their needs
- Check status on current orders:
  - Retrieve status on current orders, and estimated delivery times
- Find bicycle parts suitable for their existing bicycle:
  - Existing bicycle owners can find recommended bicycle parts and accessories based on the serial number or model number of their bicycle
  - Existing bicycle owners, can upload a picture of their bicycle or take a picture of the serial number of their bicycle to assist with the identification of their bicycle and have recommended bicycle parts

Over the last few years the customer services departments have observed an increase in calls from fraudulent customer who are asking for support for bikes that are no longer in warranty, or bikes that have not even been purchased at AdventureWorks. The department are currently relying on the experience of customer services agents to identify this. As a result, they would like to implement a system that can help the agents track in real-time who could be making a fraudulent claim.

Finally, given its global expansion, the customer service / presales chat bot needs to respond to requests for data in near real-time regardless of where the customer is located. The chatbot should also support multiple languages such as Dutch, German, French, English, Spanish, and Japanese. This work will be handled by the AI Engineers, but they have requested a platform is provided by the Data Engineer that enables them to store conversation history.

2.5 Social Media Analysis

In recent years, the marketing department at the organization have run a wide variety of twitter campaigns at various times of the year. They are keen to measure the impact of their work by tracking social media assets such as hashtags during those campaigns. They would like to have the capability of tracking any hashtag of any name.

2.6 Connected bicycle

AdventureWorks Bicycles can be equipped with an innovate built-in bicycle computer which consist of automatic locking features of the bicycle, as well as operational status. Information captured by this bicycle computer includes:

- Bicycle model, serial number and registered owner
- Bicycle location (latitude longitude)
- Current status (stationary, in motion)
- Current speed in kilometers per hours
- Bicycle Locked / Unlocked
- Bicycle parts and components information (on electrical bicycles)

First party and 3rd party applications can have access the information of the bicycle computer that must be secure and for the integration into mobile applications and real time display of location and bike ride sharing information.

Furthermore, daily summary data can be saved to flat files that include Bicycle model, serial number, registered owner and a summary of the total miles cycled per day and the average speed.

2.7 Bicycle Maintenance services

Existing bicycle owners can opt in to getting notifications on when their bicycle needs repair, based on:

- Telemetry from electrical bicycle based on sensor data
- Bicycle usage information coming from the built-in bicycle computers based on average mileage / wear and tear

This predictive maintenance scenario is a service in which bike owners can opt-in, offered as a paid service.

Finally, all services that are proposed should have a comprehensive business continuity that meets the corporate objective of minimizes restore times when recovering the data for a given service. # DP 200 - Implementing a Data Platform Solution

3 Lab 1 - Azure for the Data Engineer

Estimated Time: 60 minutes

Pre-requisites: It is assumed that the case study for this lab has already been read.

Lab files: The files for this lab are located in the *Allfiles\Labfiles\Starter\DP-200.1* folder.

3.1 Lab overview

The students will take the information gained in the lessons and from the case study to scope out the deliverables for a digital transformation project within AdventureWorks. They will first identify how the evolving use of data has presented new opportunities for the organization. The students will also explore which Azure Data Platform services can be used to address the business needs and define the tasks that will be performed by the data engineer. Finally, students will finalize the data engineering deliverables for AdventureWorks.

3.2 Lab objectives

After completing this lab, you will be able to:

1. Identify the evolving world of data within AdventureWorks.
2. Determine the Azure Data Platform services to use for AdventureWorks.
3. Identify the tasks to be performed by the Data Engineer.
4. Finalize the data engineering deliverables for AdventureWorks.

3.3 Scenario

You have been hired as a Senior Data Engineer to advise the management and employees of AdventureWorks on a digital transformation project. AdventureWorks has been selling bicycles and bicycle parts directly to end-consumer and distributors for over a decade. In the last few years, they have observed how different industries have been taking advantage of recent developments in cloud technology to offer customers a more personalized and engaging service. They want to lead the way in their industry.

You have been asked to work with the IT department to perform a discovery workshop with the organization to identify which data platform technologies can be used to the benefit of both AdventureWorks and their customers. They are specifically wanting to make sure that the technology brings the business closer to their customers on a variety of levels. This can include making personalized offers when making purchases or using customer services, to offering telemetry information on the bikes that they use. There is also a requirement to manage an existing reporting system that is held in a data warehouse.

The first step is to assess the current state of the cloud technologies and make sure that they can perform the project by performing the following analysis:

1. Identified the evolving world of data within AdventureWorks
2. Determined the Azure Data Platform services to use for AdventureWorks
3. Identified the tasks to be performed by the Data Engineer
4. Finalized the data engineering deliverables for AdventureWorks

3.4 Exercise 1: Identify the evolving world of data within AdventureWorks.

Estimated Time: 15 minutes

Group exercise

The main task for this exercise is as follows:

1. From the case study, identify the data requirements of AdventureWorks and identify if the data structure for the requirement is structured, semi-structured or unstructured.
2. The instructor will discuss the findings with the group.

3.4.1 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-Lab01-Ex01.docx** from the **Allfiles\Labfiles\Starter\DP-200.1** 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.

3.4.2 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 requirements and the associated data structures that meets the requirement.

3.5 Exercise 2: Determine the Azure Data Platform services to use for AdventureWorks

Estimated Time: 15 minutes

Group exercise

The main task for this exercise are as follows:

1. Determine the data platform technology that delivers the identified data requirements.
2. The instructor will discuss the findings with the group.

3.5.1 Task 1: Determine the data platform technology that delivers the identified data requirements

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab01-Ex02.docx** from the **Allfiles\Labfiles\Starter\DP-200.1** folder.
2. As a group, spend **10 minutes** discussing and listing the data platform technologies against a given data requirement that your group has identified within the case study document.

3.5.2 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 requirements and the associated data platform technology that meets the requirement.

3.6 Exercise 3: Identify the tasks to be performed by the Data Engineer

Estimated Time: 15 minutes

Individual exercise

The main task for this exercise are as follows:

1. Select one of the requirements and determine the high-level tasks that will perform to meet the data requirement selected.
2. The instructor will discuss the findings with the group.

3.6.1 Task 1: Determine the high-level tasks that you think you will perform to meet the data requirement.

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab01-Ex03.docx** from the **Allfiles\Labfiles\Starter\DP-200.1** folder.
2. Individually, spend **10 minutes** listing the high-level tasks that you think you will perform to meet the data requirement you have selected.

3.6.2 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 data requirements and the high-level tasks that you will perform to meet the data requirement you have selected.

3.7 Exercise 4: Finalize the data engineering deliverables for AdventureWorks.

Estimated Time: 15 minutes

Individual exercise

The main task for this exercise are as follows:

1. Finalize the data engineering deliverables for AdventureWorks
2. The instructor will discuss the findings with the group.

3.7.1 Task 1: Finalize the data engineering deliverables for AdventureWorks

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab01-Ex04.docx** from the **Allfiles\Labfiles\Starter\DP-200.1** folder.
2. Individually, spend **10 minutes** listing the data engineering deliverables for AdventureWorks as reflected by the next five modules of this course.

3.7.2 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 the data engineering deliverables for AdventureWorks as reflected by the next five modules of this course. # DP 200 - Implementing a Data Platform Solution

4 Lab 2 - Working with Data Storage

Estimated Time: 60 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-200.2* folder.

4.1 Lab overview

In this lab, the students will be able to determine the appropriate storage type to implement against a given set of business and technical requirements. They will be able to create Azure storage accounts and Data Lake Storage account and explain the difference between Data Lake Storage version 1 and version 2. They will also be able to demonstrate how to perform data loads into the data storage of choice.

4.2 Lab objectives

After completing this lab, you will be able to:

1. Choose a data storage approach in Azure
2. Create an Azure Storage Account
3. Explain Azure Data Lake Storage
4. Upload data into Azure Data Lake

4.3 Scenario

You have been hired as a Senior Data Engineer to implement a technology solution that is part of a digital transformation project. The organization is migrating an Internet Information Services (IIS) that hosts the company website to Azure. The developers are in the process of transferring the web application and its logic to Azure Web Apps and they have asked you to prepare a data store for them that can be used to host the static images that are used on the website.

In addition, the information services department have informed you that their team is expanding and that they will soon be joined by data scientists that will start the process of building a predictive analytics solution. You have been asked to set up a solution that will be used to host the production environment of their work. In the first instance, you will assess what is the appropriate storage tier to create for the solution.

At the end of this work, you will have:

1. Chosen a data storage approach in Azure
2. Created an Azure Storage Account
3. Explained Azure Data Lake Storage
4. Uploaded data into Azure Data Lake

> **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.

4.4 Exercise 1: Choose a data storage approach in Azure

Estimated Time: 15 minutes

Individual exercise

The main task for this exercise are as follows:

1. From the case study, identify the data storage requirements for the static images for the website, and for the predictive analytics solution.
2. The instructor will discuss the findings with the group.

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

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab02-Ex01.docx** from the **Allfiles\Labfiles\Starter\DP-200.2** folder.

2. Spend **10 minutes** documenting the data storage requirements as outlined in the scenario of this lab. You can also use the case study document for additional reference.

4.4.2 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 two tables of data storage requirements.

4.5 Exercise 2: Create an Azure Storage Account

Estimated Time: 20 minutes

Individual exercise

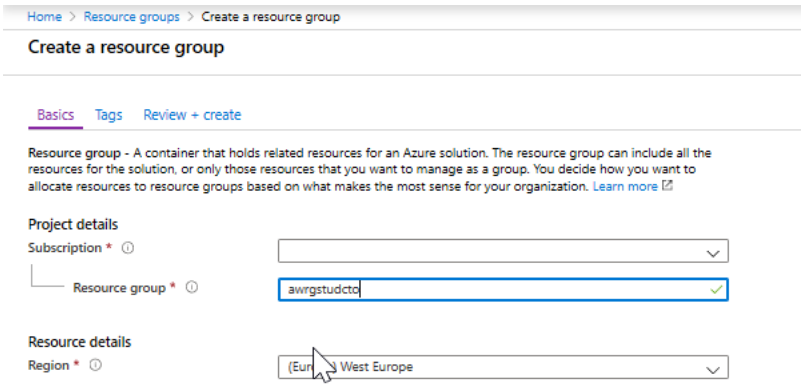
The main tasks for this exercise are as follows:

1. Create Azure resource group named **awrgstudxx** in the region closest to the lab location, where **xx** are your initials.
2. Create and configure a storage account named **awsastudxx** in the region closest to the lab location within the resource group awrgstudxx, where **xx** are your initials.
3. Create a container named **images**, **phonecalls** and **tweets** within the awsastudxx storage account.
4. Upload some graphics to the images container of the storage account.

4.5.1 Task 1: Create and configure a resource group.

1. From the lab virtual machine, start Microsoft Edge, browse to the Azure portal at **http://portal.azure.com** and sign in by using the account that has been assigned to you for the course.
2. In the Azure portal, click on the **Resource groups** icon.
3. In the **Resource groups** screen, click on **+ Add** to create the first resource group with the following settings:
   - **Subscription**: the name of the subscription you are using in this lab
   - **Resource group name**: **awrgstudxx**, where **xx** are your initials.
   - **Resource group location**: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs.

     Note: To identify Azure regions available in your subscription, refer to **https://azure.microsoft.com/en-us/regions/offers/**

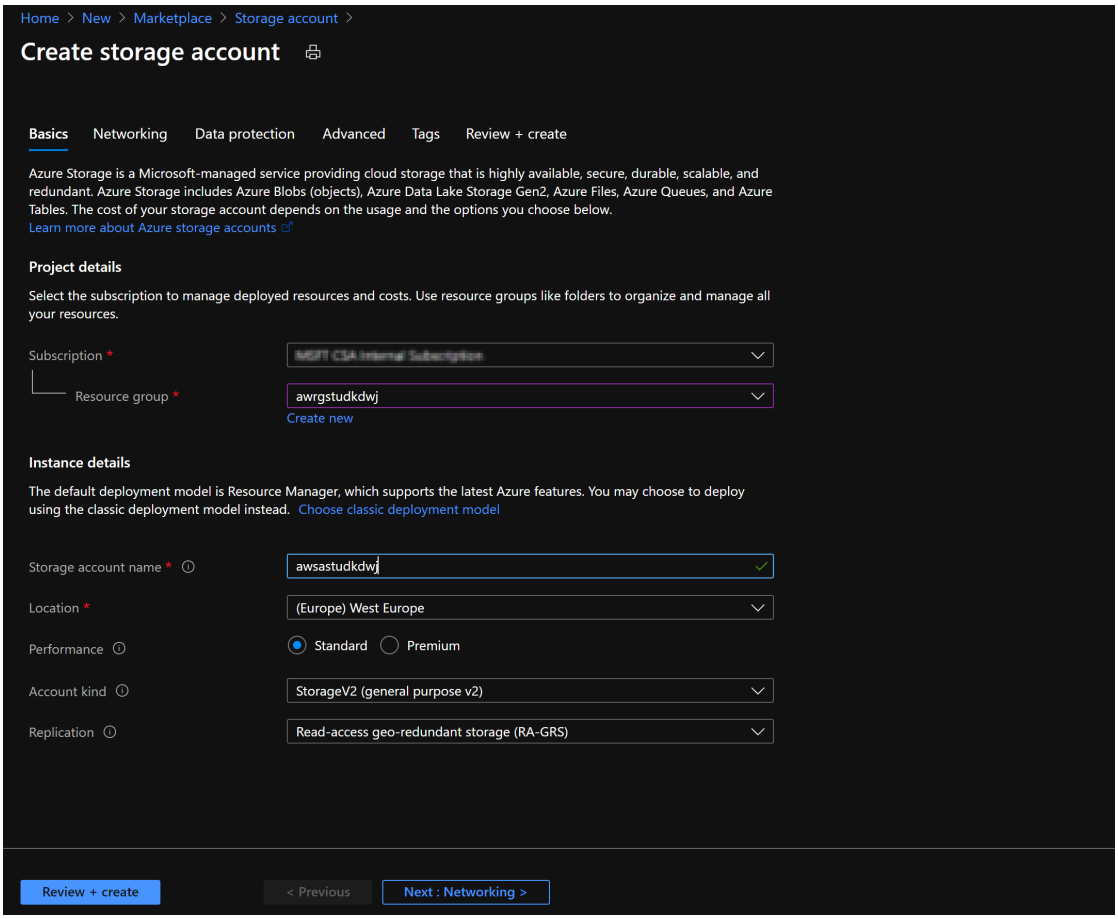


4. In the Create a resource group screen, click on **Review + Create**.
5. In the Create a resource group screen, click on **Create**.

Note: it will take approximately 30 seconds to create a resource group. You can check the notifications area to check when the creation in complete.

4.5.2 Task 2: Create and configure a storage account.

1. In the Azure portal, at the top left of the screen, click on the **Home** hyperlink
2. In the Azure portal, click on the **+ Create a resource** icon.
3. In the New screen, click in the **Search the Marketplace** text box, and type the word **storage acount**. Click **Storage account** in the list that appears.
4. In the **Storage account** screen, click **Create**.
5. From the **Create storage account** screen, create the first storage account with the following settings:
   - Under the project details, specify the following settings:
     - **Subscription**: the name of the subscription you are using in this lab
     - **Resource group**: **awrgstudxx**, where **xx** are your initials.
   - Under the instance details, specify the following settings:
     - **Storage account name**: **awsastudxx**, where **xx** are your initials.
     - **Location**: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs.
     - **Performance**: **Standard**.
     - **Account kind**: **StorageV2 (general purpose v2)**.
     - **Replication**: **Read-access geo-redundant storage (RA_GRS)**

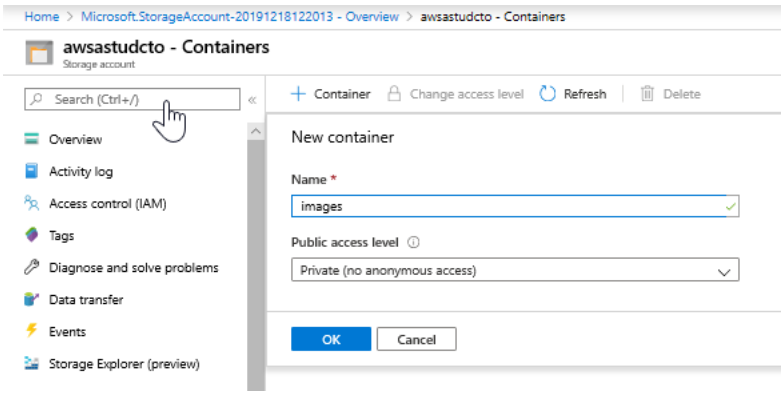


6. In the **Create storage account** screen, click **Review + create**.
7. After the validation of the **Create storage account*** screen, click **Create**.

Note: The creation of the storage account will take approximately 90 seconds while it provisions the disks and the configuration of the disks as per the settings you have defined.

4.5.3 Task 3: Create and configure a container within the storage account.

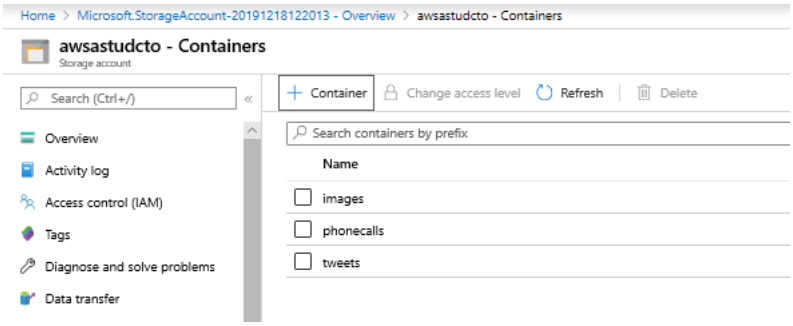
1. In the Azure portal, a message states that *Your deployment is complete*, click on the button **Go to resource**.
2. In the **awsastudxx** screen, where **xx** are your initials, under the **Blob Service** click **Containers**.
3. In the **awsastudxx - Containers** screen, at the top left, click on the **+ Container** button.
4. From the **New Container*** screen, create a container with the following settings:
   - Name: **images**.
   - Public access level: **Private (no anonymous access)**



5. In the **New Container** screen, click **Create**.

   Note: The creation of the container is immediate and will appear in the list of the **awrgstudxx - Containers** screen.

6. Repeat steps 4 -5 to create a container named **phonecalls** with the public access level of **Private (no anonymous access)**
7. Repeat steps 4 -5 to create a container named **tweets** with the public access level of **Private (no anonymous access)**. Your screen should look as the graphic below:



4.5.4 Task 4: Upload some graphics to the images container of the storage account.

1. In the Azure portal, in the **awsastudxx - Containers** screen, click on the **images** item in the list.
2. In the **images** screen, click on the **Upload** button.
3. In the **Upload blob** screen, in the Files text box, click on the **folder** icon to the right of the text box.
4. In the **Open** dialog box, browse to **Labfiles\Starter\DP-200.2\website graphics** folder. Highlight the following files:
   - one.png
   - two.png
   - three.png

- No.png

5. In the **Open** dialog box, click **Open**.
6. In the **Upload blob** screen, click on the **Upload** button.
7. Close the **Upload blob** screen, and close the **images** screen.
8. Close the **awsastudxx - Containers** screen, and in the Azure portal, navigate to the **Home** screen.

   Note: The upload of the files will take approximately 5 seconds. Once completed, they will appear in a list in the upload blobs screen.

   Result: After you completed this exercise, you have created a Storage account named awsastudxx that has a container named images that contains four graphics files that are ready to be used on the AdventureWorks website.

4.6 Exercise 3: Explain Azure Data Lake Storage

Estimated Time: 15 minutes

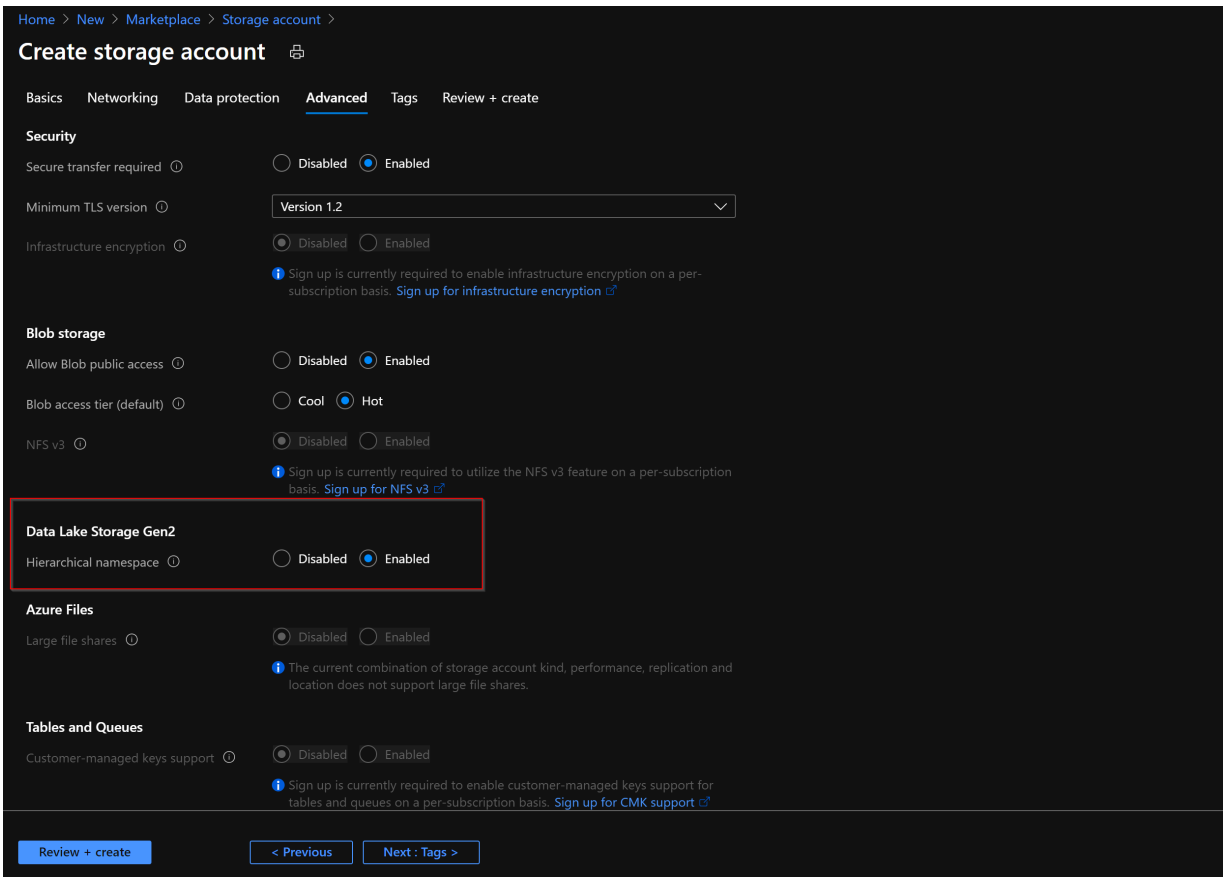
Individual exercise

The main tasks for this exercise are as follows:

1. Create and configure a storage account named **awdlsstudxx** as a Data Lake Store Gen2 storage type in the region closest to the lab location, within the resource group awrgstudxx, where **xx** are your initials.
2. Create containers named **logs** and **data** within the awdlsstudxx storage account.

4.6.1 Task 1: Create and configure a storage account as a Data Lake Store Gen II store.

1. In the Azure portal, click on **+ Create a resource** icon.
2. In the New screen, click in the **Search the Marketplace** text box, and type the word **storage**. Click **Storage account** in the list that appears.
3. In the **Storage account** blade, click **Create**.
4. From the **Create storage account*** blade, create a storage account with the following settings:
   - Under the project details, specify the following settings:
     - **Subscription**: the name of the subscription you are using in this lab
     - **Resource group name**: **awrgstudxx**, where **xx** are your initials.
   - Under the instance details, specify the following settings:
     - **Storage account name**: **awdlsstudxx**, where **xx** are your initials.
     - **Location**: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs.
     - **Performance**: **Standard**.
     - **Account kind**: **StorageV2 (general purpose v2)**.
     - **Replication**: **Read-access geo-redundant storage (RA_GRS)**
5. Click on the **Advanced** tab.
6. Under Data Lake Storage Gen2, click **Enabled** under **Hierarchical namespace**.



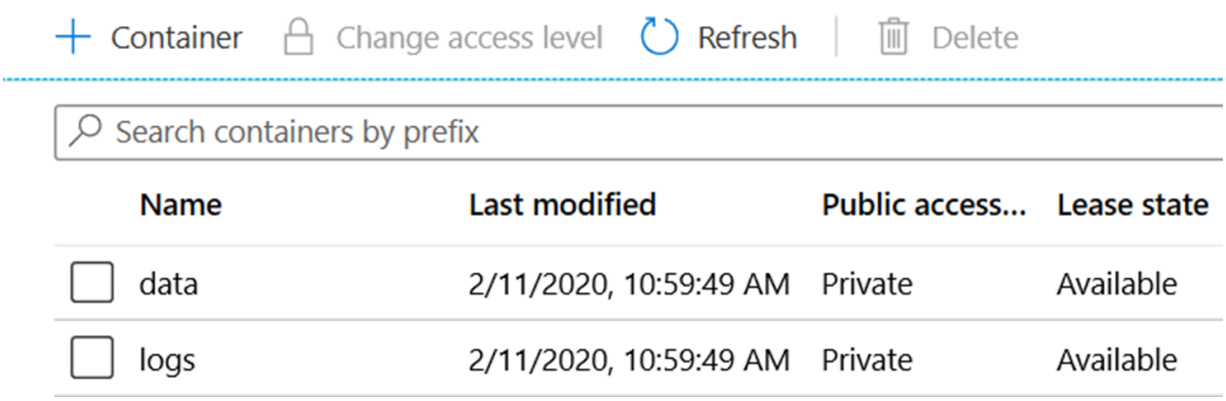
7. In the **Create storage account** blade, click **Review + create**.
8. After the validation of the **Create storage account*** blade, click **Create**.

   Note: The creation of the storage account will take approximately 90 seconds while it provisions the disks and the configuration of the disks as per the settings you have defined.

4.6.2 Task 2: Create and configure a Container within the storage account.

1. In the Azure portal, a message states that *Your deployment is complete*, click on the button **Go to resource**.
2. In the **awdlsstudxx** screen, where **xx** are your initials, click **Containers**.
3. In the **awrgstudxx - Containers** screen, at the top left, click on the **+ Containers** button.
4. From the **New** screen, create two containers with the following name:
   - Name: **data** with the public access level of **Private (no anonymous access)**.
   - Name: **logs** with the public access level of **Private (no anonymous access)**.
5. In the **New Containers** screen, click **Create**.

   Note: The creation of the file system is immediate and will appear in the list of the **awdlsstudxx - Containers** screen as follows.



Result: After you completed this exercise, you have created a Data Lake Gen2 Storage account named awdlsstudxx that has a file system named data and logs.

4.7 Exercise 4: Upload data into Azure Data Lake.

Estimated Time: 10 minutes

Individual exercise

The main tasks for this exercise are as follows:

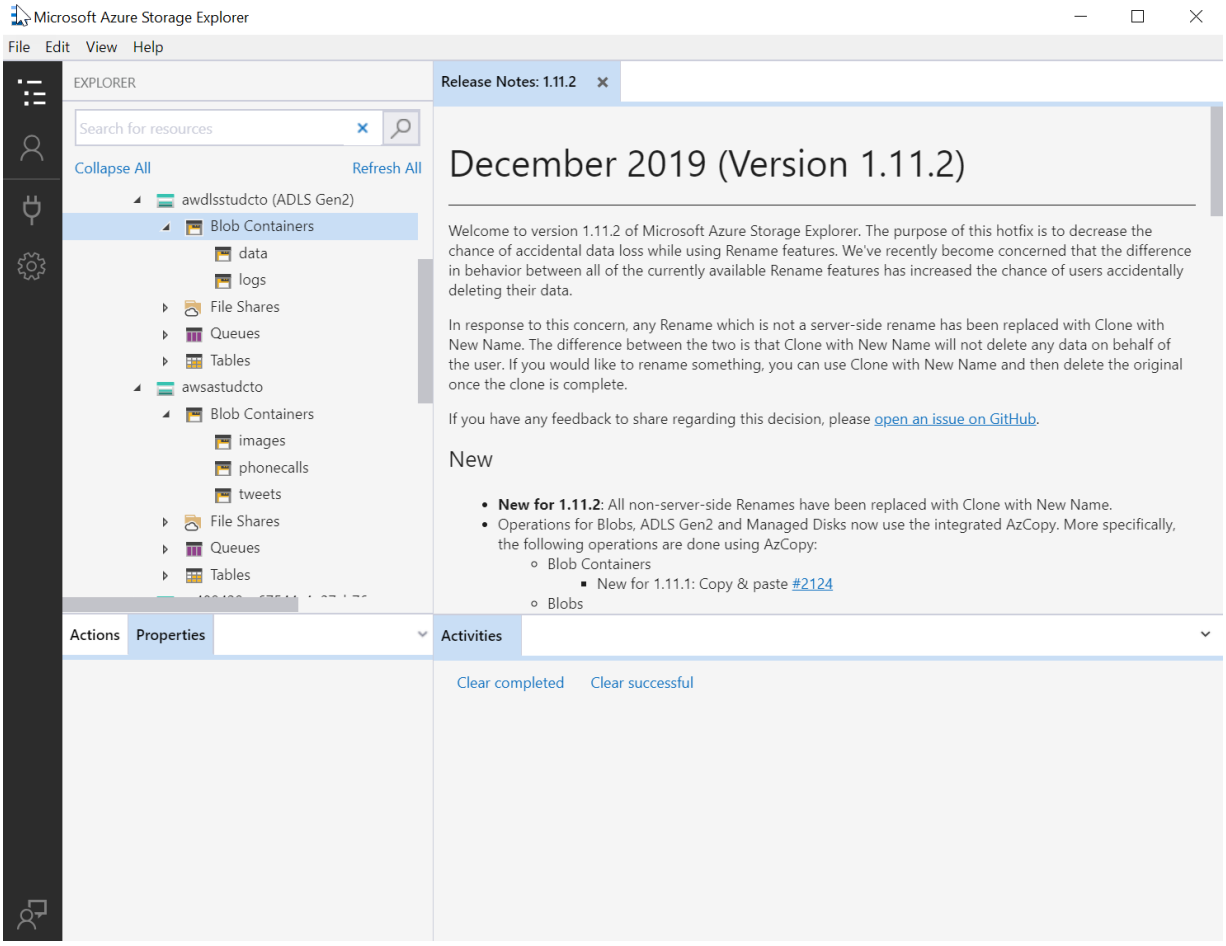
1. Install and start Microsoft Azure Storage Explorer
2. Upload some data files to the containers of the Data Lake Gen II Storage Account.

4.7.1 Task 1: Install Storage Explorer.

3. In the Azure portal, in the **awdlsstudxx** overview page, navigate to **Open in Explorer** and then click on the **Download Azure Storage Explorer** hyperlink.
4. You are taken to the following web page for Azure Storage Explorer where there is a button that states **Download now**. click on this button.
5. In the Microsoft Edge dialog box click **Save**, when the download is complete, click on **View downloads**, in the download screen in Microsoft Edge, click on **Open folder**. This will open the Downloads folder.
6. Double click the file **StorageExplorer.exe**, in the User Account Control dialog box click on **Yes**.
7. In the License Agreement screen, select the radio button next to **I agree the agreement**, and then click on **Install**.

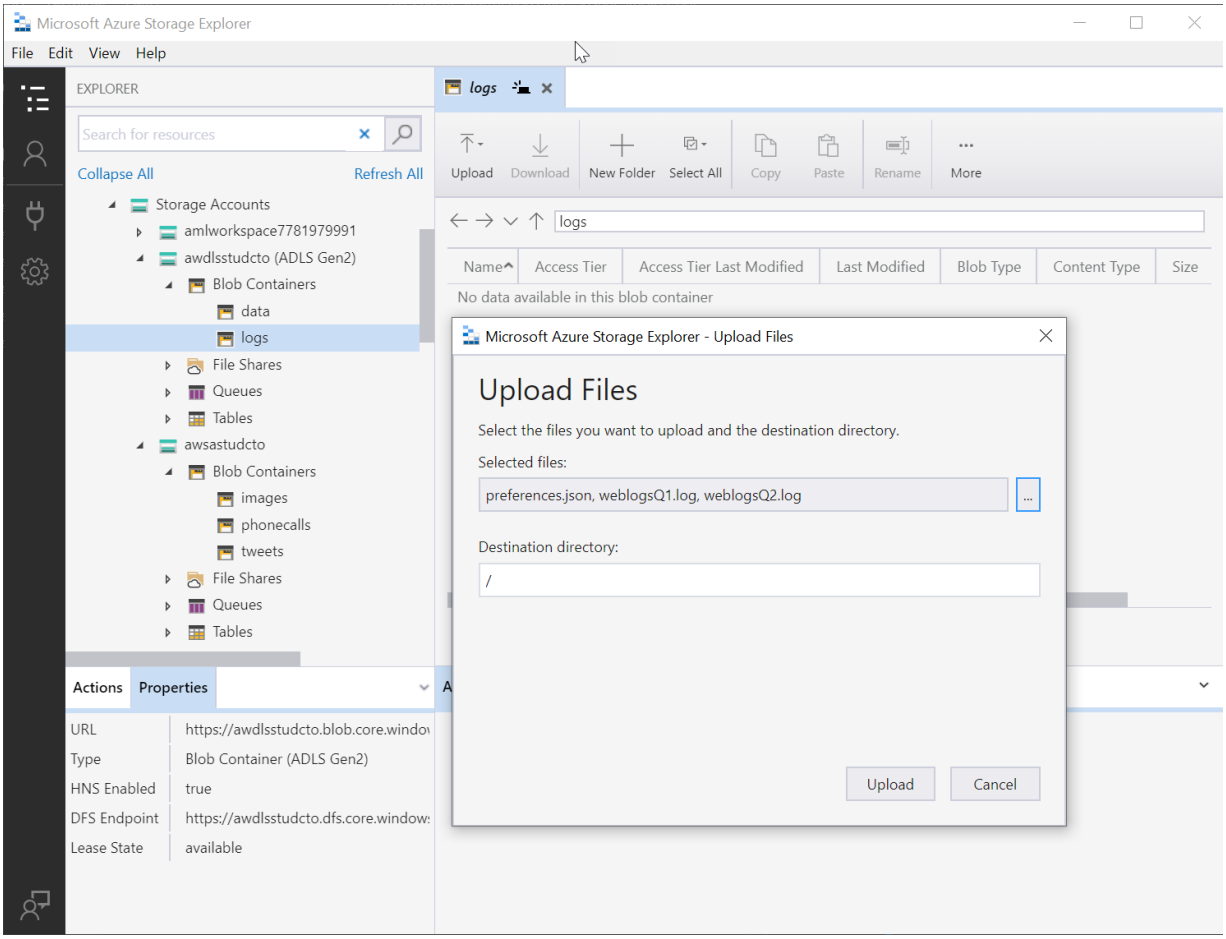
   Note: The installation of Storage Explorer can take approximately 4 minutes. Azure Storage Explorer allows you to easily manage the contents of your storage account with Azure Storage Explorer. Upload, download, and manage blobs, files, queues, tables, and Cosmos DB entities. It also enables you to gain easy access to manage your virtual machine disks.

8. On completion of the installation, ensure that the checkbox next to **Launch Microsoft Azure Storage Explorer** is selected and then click **Finish**. Microsoft Azure Storage Explorer opens up and lists your subscriptions.
9. In Storage Explorer, select **Manage Accounts** to go to the **Account Management Panel**.
10. The left pane now displays all the Azure accounts you've signed in to. To connect to another account, select **Add an account**
11. If you want to sign into a national cloud or an Azure Stack, click on the Azure environment dropdown to select which Azure cloud you want to use. Once you have chosen your environment, click the **Sign in...** button.
12. After you successfully sign in with an Azure account, the account and the Azure subscriptions associated with that account are added to the left pane. Select the Azure subscriptions that you want to work with, and then select **Apply**. The left pane displays the storage accounts associated with the selected Azure subscriptions.



4.7.2 Task 2: Upload data files to the data and logs container of the Data Lake Gen II Storage Account.

1. In Azure Storage Explorer, click on the arrow to expand your subscription.
2. Under **Storage Accounts**, search for the storage account **awdlsstudxx (ADLS Gen2)**, and click on the arrow to expand it.
3. Under **Blob Containers**, click on the arrow to expand it and show the **logs** file system. Click on the **logs** file system.
4. In Azure Storage Explorer, click on the arrow next to the **Upload** icon, and click on the **Upload Files...**.
5. In Upload Files dialog box, click on the ellipsis next to the **Selected files** text box.
6. In the **Choose files to upload** dialog box, browse to **Labfiles\Starter\DP-200.2\logs** folder. Highlight the following files:
   - weblogsQ1.log
   - weblogsQ2.log
   - preferences.json
7. In the **Choose files to upload** dialog box, click **Open**.
8. In the **Upload Files** screen, click on the **Upload** button.



9. Under **Blob Containers**, click on the arrow to expand it and show the **data** file system. Click on the **data** file system.
10. In Azure Storage Explorer, click on the arrow next to the **Upload** icon, and click on the **Upload Files...**.
11. In Upload Files dialog box, click on the ellipsis next to the **Selected files** text box.
12. In the **Choose files to upload** dialog box, browse to **Labfiles\Starter\DP-200.2\Static Files** folder. Highlight the following files:
    - DimDate2.txt
13. In the **Choose files to upload** dialog box, click **Open**.
14. In the **Upload Files** screen, click on the **Upload** button.
15. Repeat the steps to upload the preferences.JSON file from the **Labfiles\Starter\DP-200.2\logs** folder to the **data** file system in the Data Lake Store gen2

    Note: The upload of the files will take approximately 5 seconds. You will see a message in Azure Storage Explorer that states **Your view may be out of data. Do you want to refresh? Click Yes**. Once completed, all two files will appear in a list in the upload blobs screen.

```
![Files uploaded to Containers in Azure Storage Explore](/home/ll/Azure_clone/Azure_new/DP-200-Implemen
```

16. In Azure Storage Explorer, in the data file system, click on the + **New Folder** button.
17. In the New Folder screen, in the New folder name text box, type **output**.
18. Close down Azure Storage Explorer.
19. Return to the Azure portal, and navigate to the **Home** blade.

    Result: After you completed this exercise, you have created a Data Lake Gen II Storage account named awdlsstudxx that has a file system named data that contains two weblog files that are ready to be used by the Data scientists at AdventureWorks. # DP 200 - Implementing a Data Platform Solution

5 Lab 3 - Enabling Team Based Data Science with Azure Databricks

Estimated Time: 75 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-200.3* folder.

5.1 Lab overview

By the end of this lab the student will be able to explain why Azure Databricks can be used to help in Data Science projects. The students will provision an Azure Databricks instance and will then create a workspace that will be used to perform simple data preparation tasks from a Data Lake Store Gen2 store. Finally, the student will perform a walk-through of performing transformations using Azure Databricks.

5.2 Lab objectives

After completing this lab, you will be able to:

1. Explain Azure Databricks
2. Work with Azure Databricks
3. Read data with Azure Databricks
4. Perform transformations with Azure Databricks

5.3 Scenario

In response to the Information Services (IS) department, you will start the process of building a predictive analytics platform by listing out the benefits of using the technology. The department will be joined by data scientists and they want to ensure that there is a predictive analytics environment available to the new team members.

You will stand up and provision an Azure Databricks environment, and then test that this environment works by performing a simple data preparation routine on the service by ingesting data from a pre-existing Data Lake Storage Gen2 account. As a data engineer, it has been indicated to you that you may be required to help the data scientists perform data preparation exercises. To that end, you have been recommended to walk-through a notebook that can help you perform basic transformations.

At the end of this lad, you will have:

1. Explained Azure Databricks
2. Worked with Azure Databricks
3. Read data with Azure Databricks
4. Performed transformations with Azure 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.

5.4 Exercise 1: Explain Azure Databricks

> **Important**: Perform **exercise 2 first**, and return to exercise 1 after starting the creation of a Databricks Cluster in exercise 2, as it will take 10 minutes to provision.

Estimated Time: 15 minutes

Individual exercise

The main task for this exercise are as follows:

1. From the content you have learned in this course so far, identify the digital transformation requirement that Azure Databricks will meet and a candidate data source for Azure Databricks.
2. The instructor will discuss the findings with the group.

5.4.1 Task 1: Define the digital transformation and candidate data source.

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab03-Ex01.docx** from the **Allfiles\Labfiles\Starter\DP-200.3** folder.
2. Spend **10 minutes** documenting the digital transformation requirement and candidate data source as outlined in the case study and the scenario of this lab.

5.4.2 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 identifies the digital transformation requirement that Azure Databricks will meet and a candidate data source.

5.5 Exercise 2: Work with Azure Databricks

Estimated Time: 20 minutes

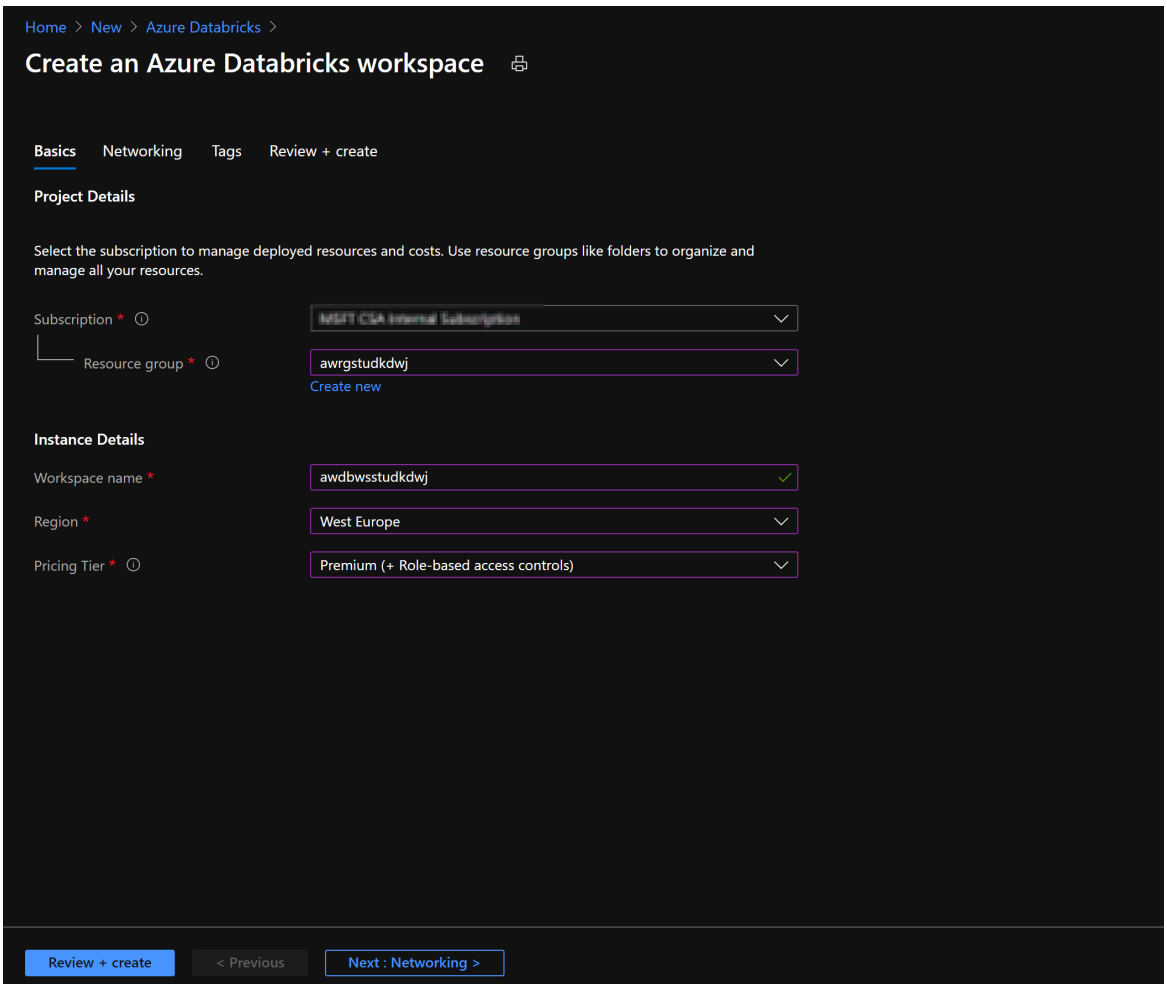
Individual exercise

The main tasks for this exercise are as follows:

1. Create an Azure Databricks Premium Tier instance in a resource group.
2. Open Azure Databricks
3. Launch a Databricks Workspace and create a Spark Cluster

5.5.1 Task 1: Create and configure an Azure Databricks instance.

1. In the Azure portal, at the top left of the screen, click on the **Home** hyperlink.
2. In the Azure portal, click on the **+ Create a resource** icon.
3. In the New screen, click in the **Search the Marketplace** text box, and type the word **databricks**. Click **Azure Databricks** in the list that appears.
4. In the **Azure Databricks** blade, click **Create**.
5. In the **Azure Databricks Service** blade, create an Azure Databricks Workspace with the following settings:
   - **Workspace name**: **awdbwsstudxx**, where **xx** are your initials.
   - **Subscription**: the name of the subscription you are using in this lab
   - **Resource group**: **awrgstudxx**, where **xx** are your initials.
   - **Location**: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs.
   - **Pricing Tier**: **Premium (+ Role-based access controls)**.

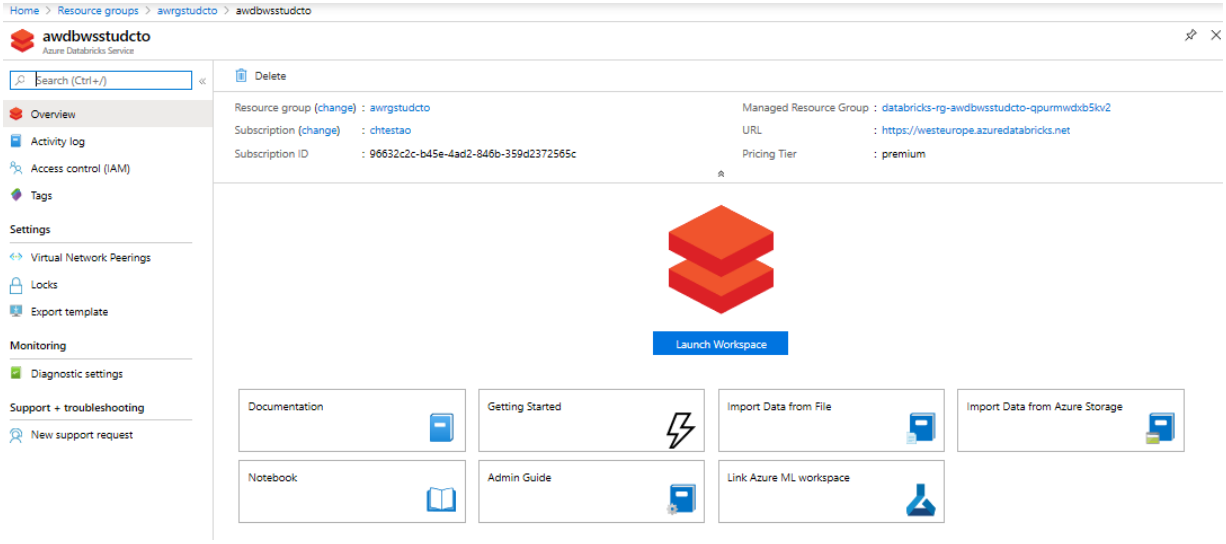


6. In the **Azure Databricks Service** blade, click **Create**.

   Note: The provision will take approximately 3 minutes. The Databricks Runtime is built on top of Apache Spark and is natively built for the Azure cloud. Azure Databricks completely abstracts out the infrastructure complexity and the need for specialized expertise to set up and configure your data infrastructure. For data engineers, who care about the performance of production jobs, Azure Databricks provides a Spark engine that is faster and performant through various optimizations at the I/O layer and processing layer (Databricks I/O).

5.5.2 Task 2: Open Azure Databricks.

1. Confirm that the Azure Databricks service has been created.
2. In the Azure portal, navigate to the **Resource group** screen.
3. In the Resource groups screen, click on the ****awrgstudxx** resource group, where **xx** are your initials.
4. In the **awrgstudxx** screen, click **awdbwsstudxx**, where **xx** are your initials to open Azure Databricks. This will open your Azure Databricks service.



5.5.3 Task 3: Launch a Databricks Workspace and create a Spark Cluster.

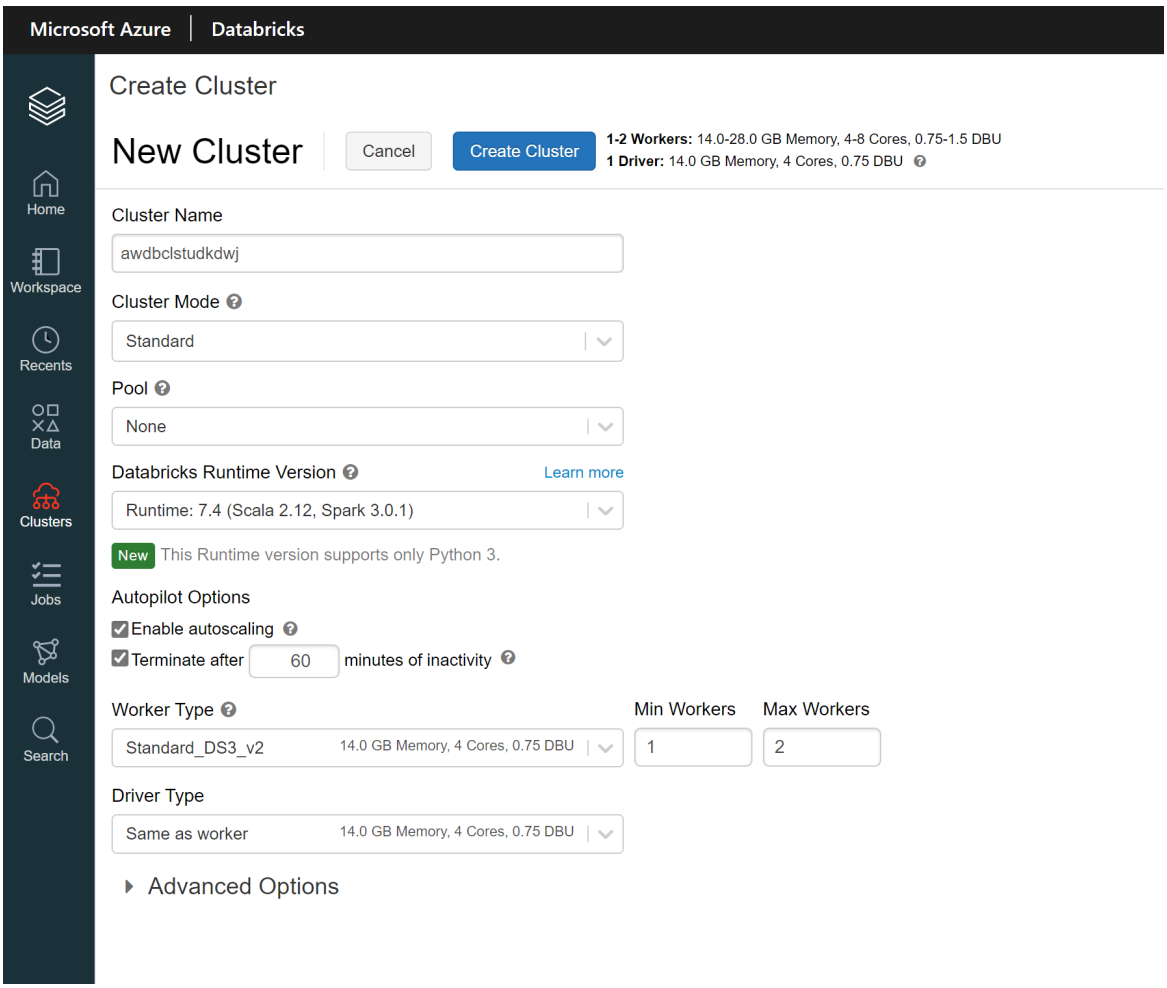
1. In the Azure portal, in the **awdbwsstudxx** screen, click on the button **Launch Workspace**.

   Note: You will be signed into the Azure Databricks Workspace in a separate tab in Microsoft Edge.

2. Under **Common Tasks**, click **New Cluster**.

3. In the **Create Cluster** screen, under New Cluster, create a Databricks Cluster with the following settings, and then click on **Create Cluster**:

   - **Cluster name**: **awdbclstudxx**, where **xx** are your initials.
   - **Cluster Mode**: **Standard**
   - **Pool**: **None**
   - **Databricks Runtime Version**: **Runtime: 7.4 (Scala 2.12, Spark 3.0.1)**
   - **Python version**: **2**
   - Make sure you select the **Terminate after 60** minutes of inactivity check box. If the cluster isn't being used, provide a duration (in minutes) to terminate the cluster.
   - **Min Workers**: **1**
   - **Max Workers**: **2**
   - Leave all the remaining options to their current settings.



4. In the **Create Cluster** screen, click on **Create Cluster** and leave the Microsoft Edge screen open.

   Note: The creation of the Azure Databricks instance will take approximately 10 minutes as the creation of a Spark cluster is simplified through the graphical user interface. You will note that the **State** of **Pending** whilst the cluster is being created. This will change to **Running** when the Cluster is created.

   Note: While the cluster is being created, **go back and perform Exercise 1**.

5.6 Exercise 3: Read data with Azure Databricks

Estimated Time: 30 minutes

Individual exercise

The main tasks for this exercise are as follows:

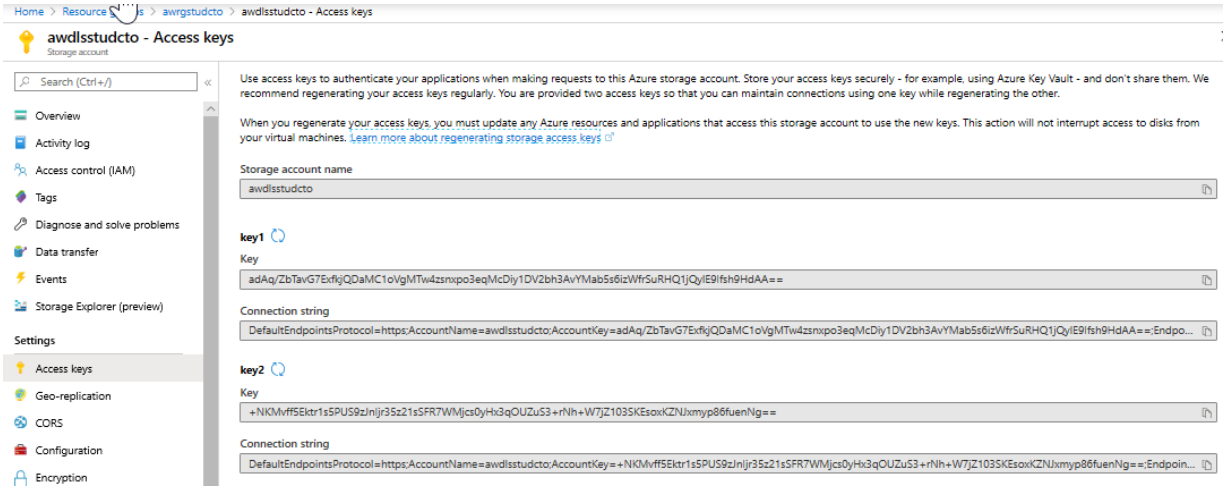
1. Confirm that the Databricks cluster has been created.
2. Collect the Azure Data Lake Store Gen2 account name
3. Enable your Databricks instance to access the Data Lake Gen2 Store.
4. Create a Databricks Notebook and connect to a Data Lake Store.
5. Read data in Azure Databricks.

5.6.1 Task 1: Confirm the creation of the Databricks cluster

1. Return back to Microsoft Edge, under **Interactive Clusters** confirm that the state column is set to **Running** for the cluster named **awdbclstudxx**, where **xx** are your initials.

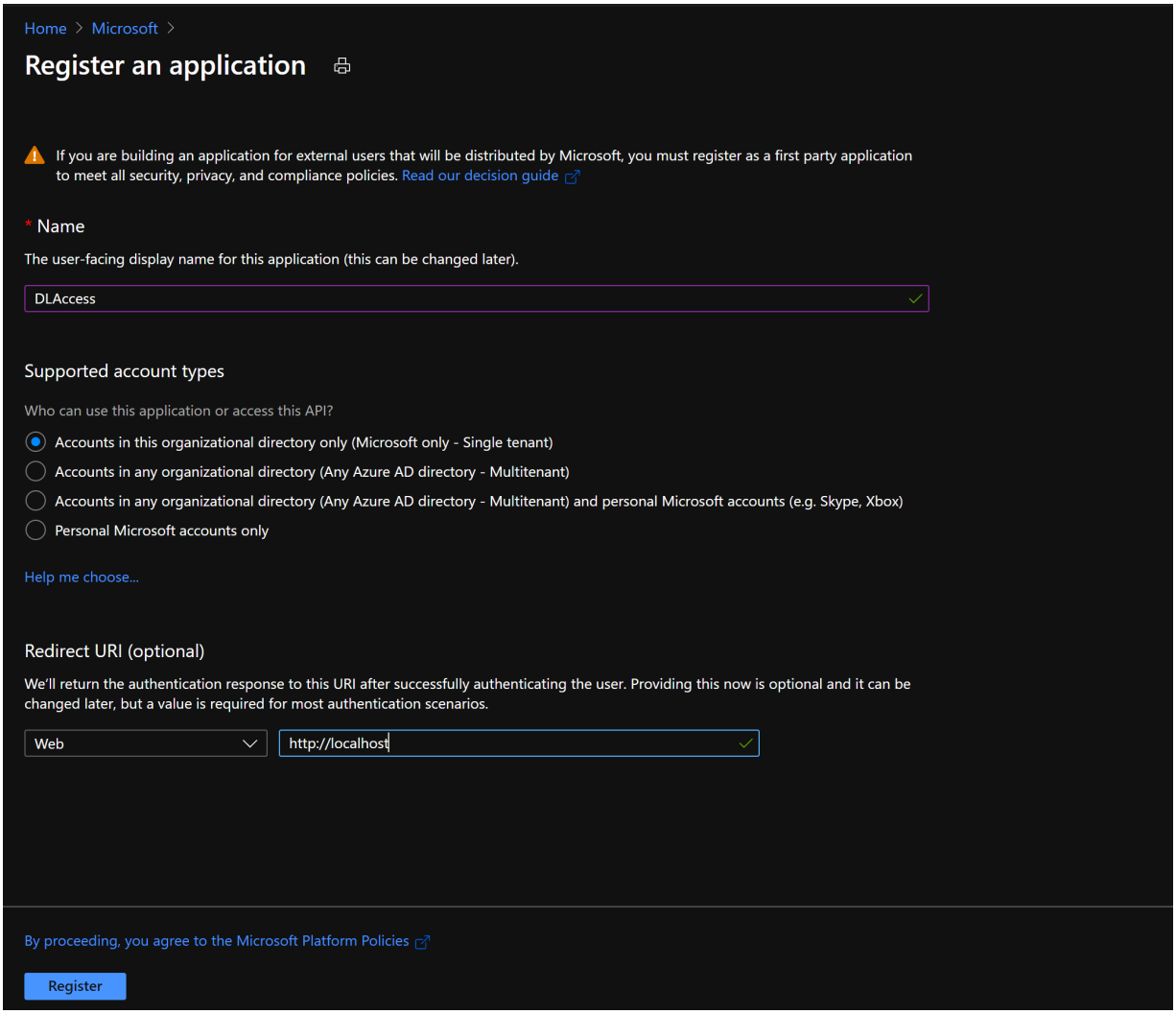
5.6.2 Task 2: Collect the Azure Data Lake Store Gen2 account name

1. In Microsoft Edge, click on the Azure portal tab, click **Resource groups**, and then click **awrgstudxx**, and then click on **awdlsstudxx**, where **xx** are your initials.
2. In the **awdlsstudxx** screen, under settings, click on **Access keys**, and then click on the copy icon next to the **Storage account name**, and paste it into Notepad.

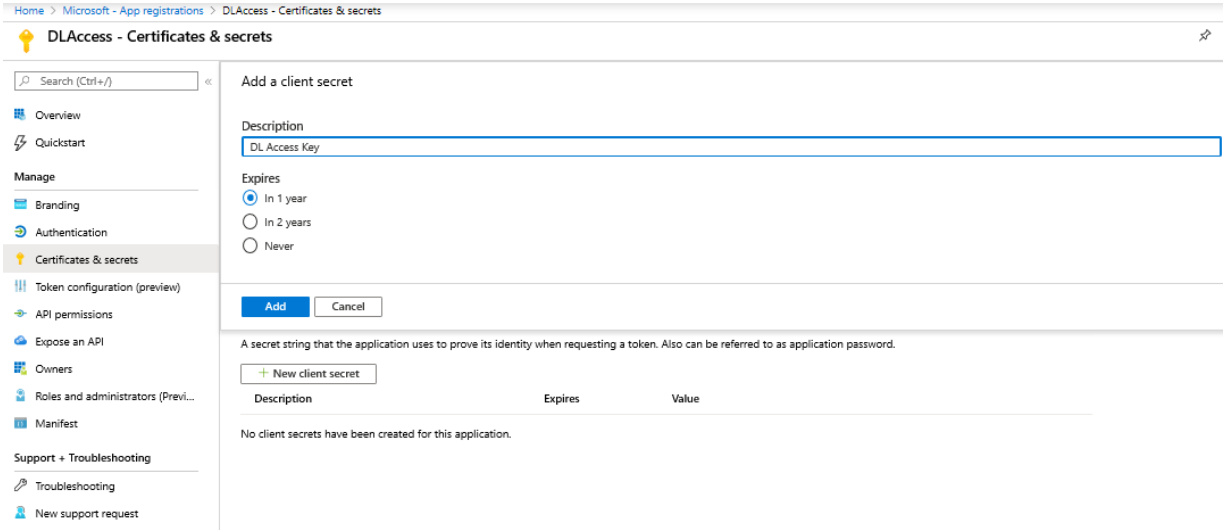


5.6.3 Task 3: Enable your Databricks instance to access the Data Lake Gen2 Store.

1. In the Azure portal, Click the **Home** hyperlink, and then click the **Azure Active Directory** icon.
2. In the **Microsoft - Overview** screen, click on **App registrations**.
3. In the **Microsoft - App registrations** screen, click on the **+ New registration** button.
4. In the register an application screen, provide the **name** of **DLAccess** and under the **Redirect URI (optional)** section, ensure **Web** is selected and type **http://localhost**for the application value. After setting the values.

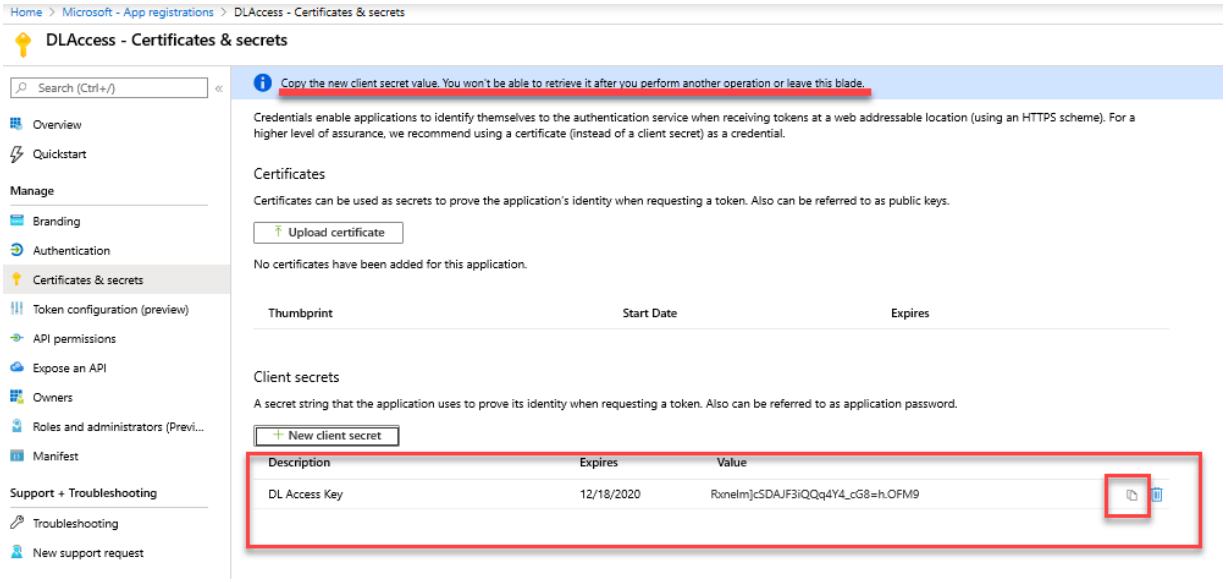


5. Click **Register**. The DLAccess screen will appear.
6. In the **DLAccess** registered app screen, copy the **Application (client) ID** and **Directory (tenant) ID** and paste both into Notepad.
7. In the **DLAccess** registered app screen, click on **Certificates and Secrets**, and the click + **New Client Secret**
8. In the Add a client secret screen. type a **description** of **DL Access Key**, and a **duration** of **In 1 year** for the key. When done, click **Add**.



Important: When you click on **Add**, the key will appear as shown in the graphic below. You

only have one opportunity to copy this key value into Notepad



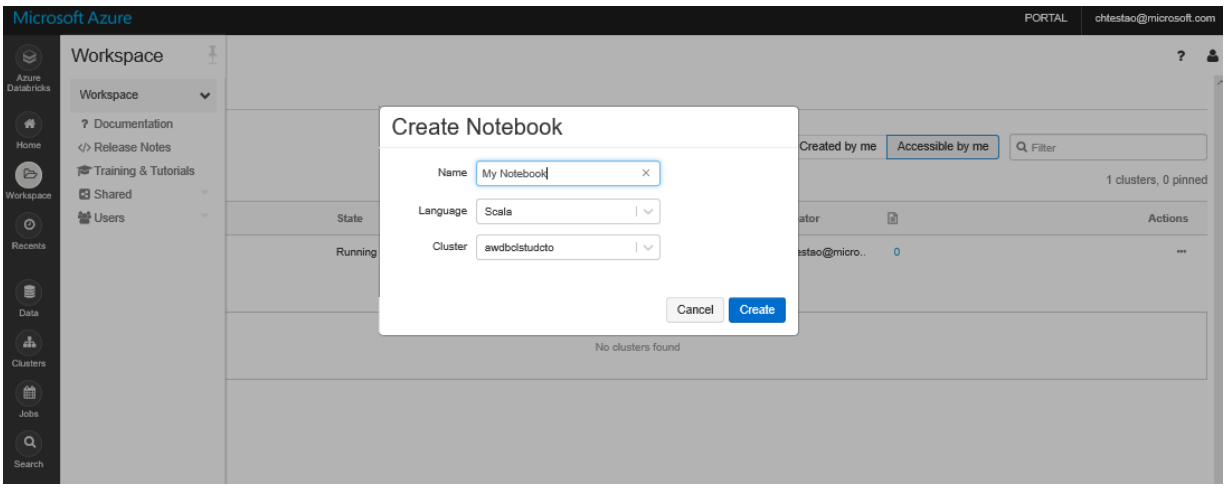
9. Copy the **Application key value** and paste it into Notepad
10. Assign the Storage Blob Data Contributor permission to your resource group. In the Azure portal, click on the **Home** hyperlink, and then the **Resource groups** icon, click on the resource group **awrgstudxx**, where **xx** are your initials.
11. In the **awrgstudxx** screen, click on **Access Control (IAM)**
12. Click on the **Role assignments** tab.
13. Click **+ Add**, and click **Add role assignment**
14. In the **Add role assignment** blade, under Role, select **Storage Blob Data Contributor**.
15. In the **Add role assignment** blade, under Select, select **DLAccess**, and then click **Save**.
16. In the Azure portal, click the **Home** hyperlink, and then click the **Azure Active Directory** icon, Note **your role**. If you have the User role, you must make sure that non-administrators can register applications.
17. Click **Users**, and then click **User settings** in the **Users - All users** blade, Check the **App registrations** setting. This value can only be set by an administrator. If set to Yes, any user in the Azure AD tenant can register an app.
18. Close down the **Users - All users** screen.
19. In the Azure Active Directory blade, click **Properties**.
20. Click on the Copy icon next to the **Directory ID** to get your tenant ID and paste this into notepad.
21. Save the notepad document in the folder **Allfiles\Labfiles\Starter\DP-200.3** as **DatabricksDetails.txt**

5.6.4 Task 4: Create a Databricks Notebook and connect to a Data Lake Store.

1. In Microsoft Edge, click on the tab **Clusters - Databricks**

   Note: You will see the Clusters page.

2. In the Azure Databricks blade on the left of Microsoft Edge, click on Under **Workspace**, click on the drop down next to **Workspace**, then point to **Create** and then click on **Notebook**.
3. In the **Create Notebook** screen, next to Name type **My Notebook**.
4. Next to the **Language** drop down list, select **Scala**.
5. Ensure that the Cluster states the name of the cluster that you have created earlier, click on **Create**



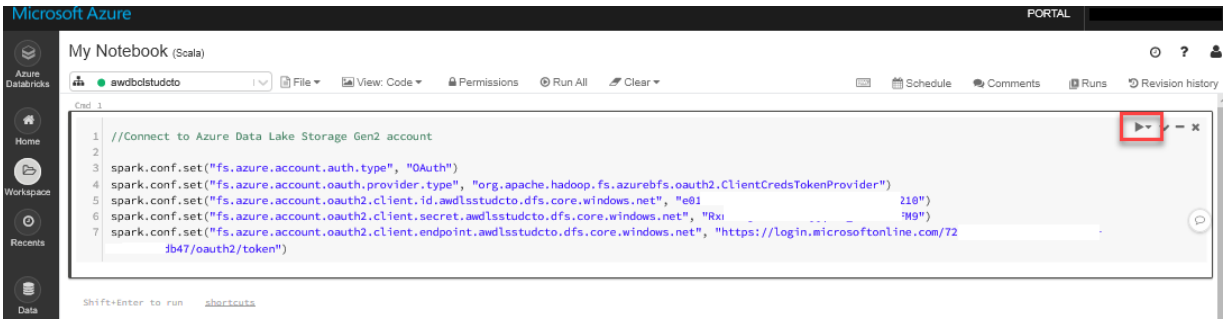
Note: This will open up a Notebook with the title My Notebook (Scala).

6. In the Notebook, in the cell **Cmd 1**, copy the following code and paste it into the cell:

```
//Connect to Azure Data Lake Storage Gen2 account

spark.conf.set("fs.azure.account.auth.type", "OAuth")
spark.conf.set("fs.azure.account.oauth.provider.type", "org.apache.hadoop.fs.azurebfs.oauth2.Clien
spark.conf.set("fs.azure.account.oauth2.client.id.<storage-account-name>.dfs.core.windows.net", "<
spark.conf.set("fs.azure.account.oauth2.client.secret.<storage-account-name>.dfs.core.windows.net"
spark.conf.set("fs.azure.account.oauth2.client.endpoint.<storage-account-name>.dfs.core.windows.ne
```

7. In this code block, replace the **application-id**, **authentication-id**, **tenant-id**, **file-system-name** and **storage-account-name** placeholder values in this code block with the values that you collected earlier and are held in notepad.

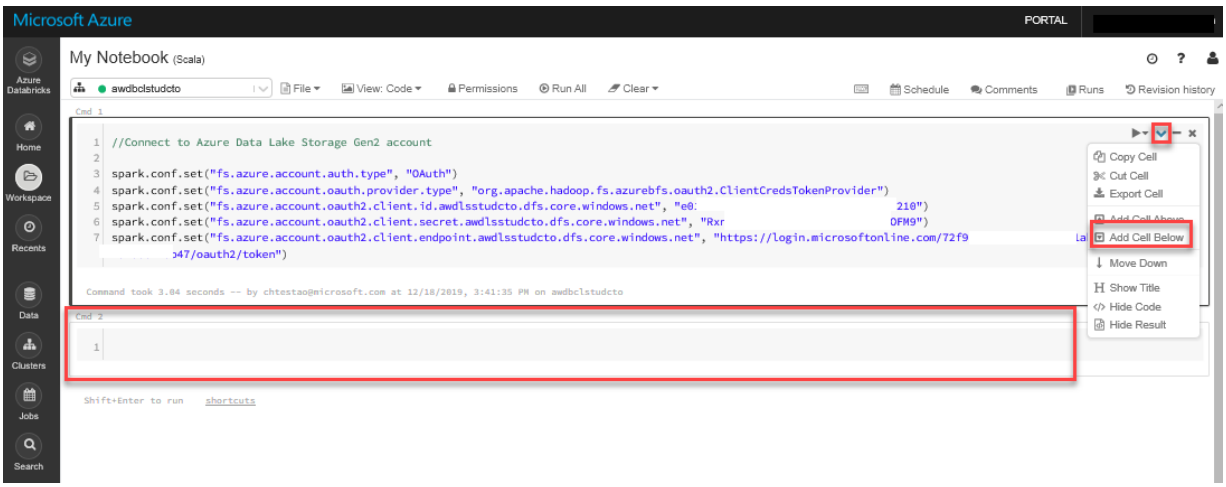
8. In the Notebook, in the cell under **Cmd 1**, click on the **Run** icon and click on **Run Cell** as highlighted in the following graphic.



Note A message will be returned at the bottom of the cell that states "Command took 0.0X seconds -- by person at 4/4/2019, 2:46:48 PM on awdbclstudxx"

5.6.5 Task 5: Read data in Azure Databricks.

1. In the Notebook, hover your mouse at the top right of cell **Cmd 1**, and click on the **Add Cell Below** icon. A new cell will appear named **Cmd2**.



2. In the Notebook, in the cell **Cmd 2**, copy the following code and paste it into the cell:

```
//Read JSON data in Azure Data Lake Storage Gen2 file system

val df = spark.read.json("abfss://<file-system-name>@<storage-account-name>.dfs.core.windows.net/p
```

3. In this code block, replace the **file-system-name** with the word **logs** and **storage-account-name** placeholder values in this code block with the value that you collected earlier and are held in notepad.

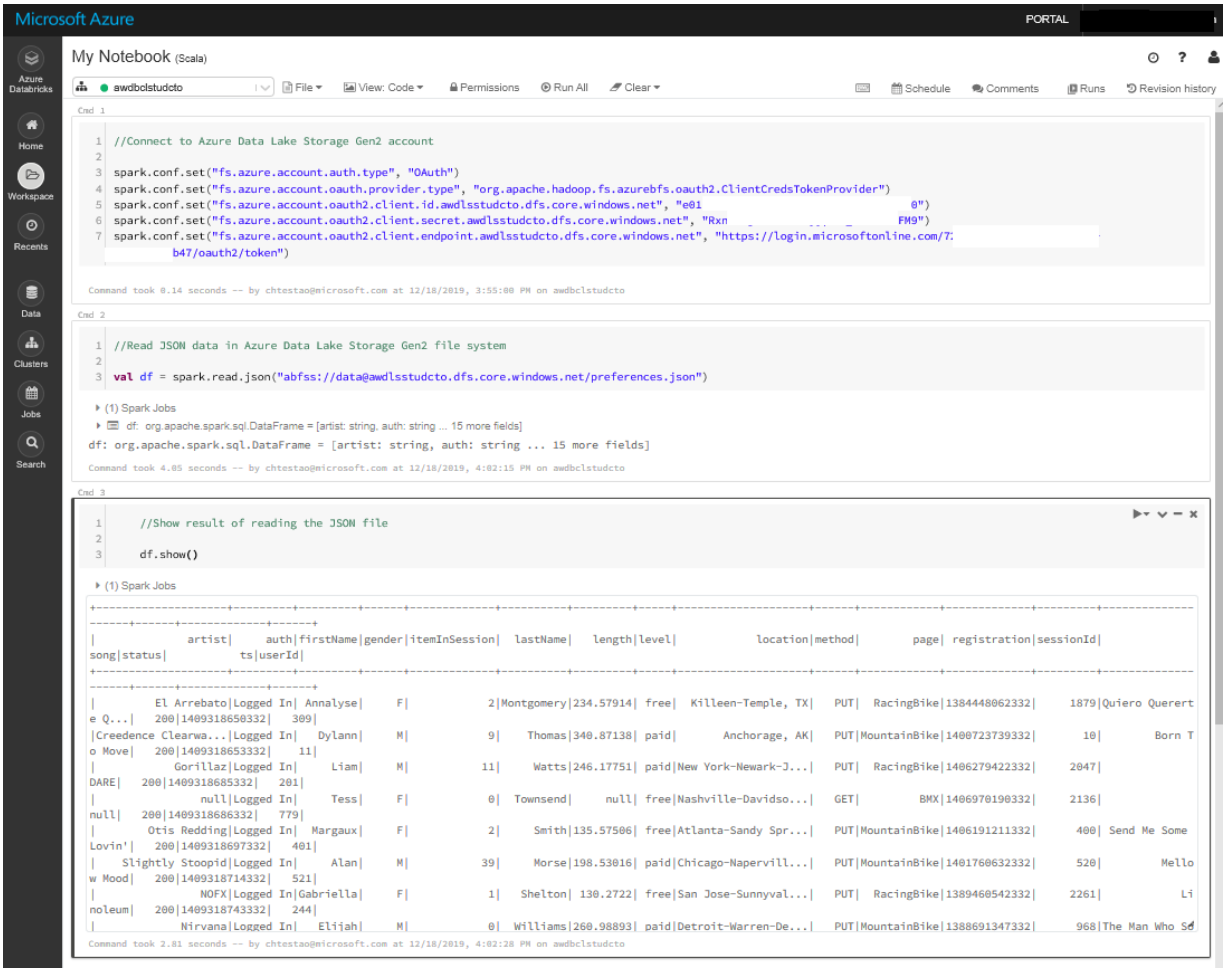
4. In the Notebook, in the cell under **Cmd 2**, click on the **Run** icon and click on **Run Cell**.

   Note A message will be returned at the bottom of the cell that states that a Spark job has executed and "Command took 0.0X seconds -- by person at 4/4/2019, 2:46:48 PM on awdb-clstudxx"

5. In the Notebook, hover your mouse at the top right of cell **Cmd 2**, and click on the **Add Cell Below** icon. A new cell will appear named **Cmd3**.

6. In the Notebook, in the cell **Cmd 3**, copy the following code and paste it into the cell:

```
//Show result of reading the JSON file

df.show()
```



7. In the Notebook, in the cell under **Cmd 3**, click on the **Run** icon and click on **Run Cell**.

   Note A message will be returned at the bottom of the cell that states that a Spark job has executed, a table of results are returned and "Command took 0.0X seconds -- by person at 4/4/2019, 2:46:48 PM on awdbclstudxx"

8. Leave the Azure Databricks Notebook open

   Result In this exercise, you have performed the necessary steps that setup up the permission for Azure Databricks to access data in an Azure Data Lake Store Gen2. You then used scala to connect up to a Data Lake Store and you read data and created a table output showing the preferences of people.

5.7 Exercise 4: Perform basic transformations with Azure Databricks

Estimated Time: 10 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Retrieve specific columns on a Dataset
2. Performing a column rename on a Dataset
3. Add an Annotation
4. If Time permits: Additional transformations

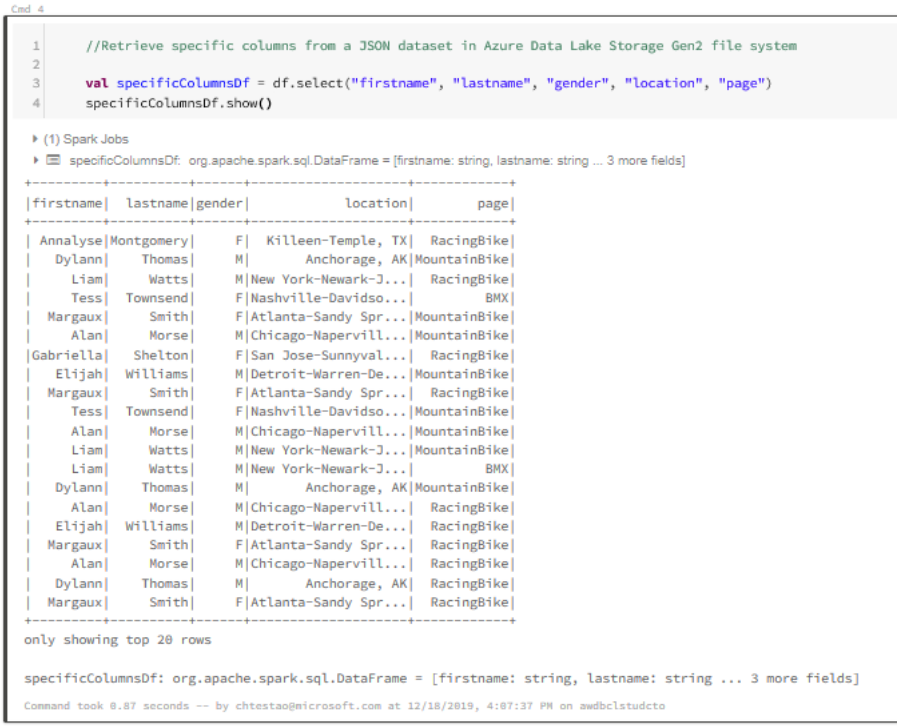
5.7.1 Task 1: Retrieve specific columns on a Dataset

1. In the Notebook, hover your mouse at the top right of cell **Cmd 3**, and click on the **Add Cell Below** icon. A new cell will appear named **Cmd4**.
2. In the Notebook, in the cell **Cmd 4**, copy the following code and paste it into the cell:

```
//Retrieve specific columns from a JSON dataset in Azure Data Lake Storage Gen2 file system

val specificColumnsDf = df.select("firstname", "lastname", "gender", "location", "page")
specificColumnsDf.show()
```

3. In the Notebook, in the cell under **Cmd 4**, click on the **Run** icon and click on **Run Cell**.

   Note A message will be returned at the bottom of the cell that states that a Spark job has executed, a table of results are returned and "Command took 0.0X seconds -- by person at 4/4/2019, 2:46:48 PM on awdbclstudxx"



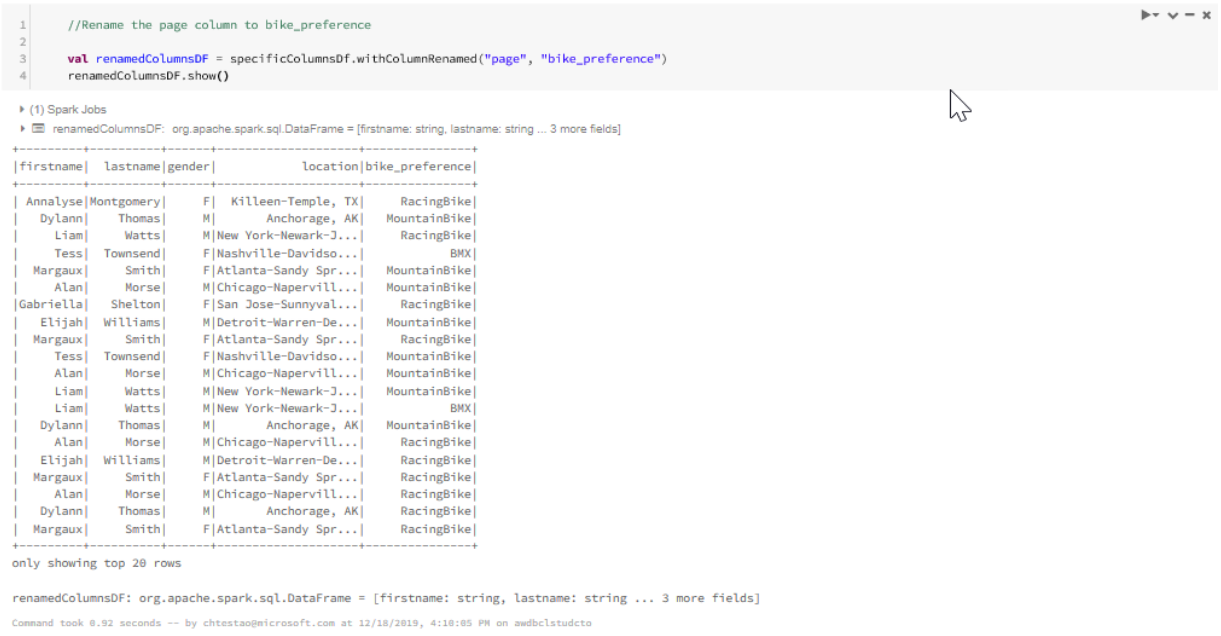
5.7.2 Task 2: Performing a column rename on a Dataset

1. In the Notebook, hover your mouse at the top right of cell **Cmd 4**, and click on the **Add Cell Below** icon. A new cell will appear named **Cmd5**.
2. In the Notebook, in the cell **Cmd 5**, copy the following code and paste it into the cell:

```
//Rename the page column to bike_preference

val renamedColumnsDF = specificColumnsDf.withColumnRenamed("page", "bike_preference")
renamedColumnsDF.show()
```

3. In the Notebook, in the cell under **Cmd 5**, click on the **Run** icon and click on **Run Cell**.

   Note A message will be returned at the bottom of the cell that states that a Spark job has executed, a table of results are returned and "Command took 0.0X seconds -- by person at 4/4/2019, 2:46:48 PM on awdbclstudxx"



5.7.3 Task 3: Adding Annotations

1. In the Notebook, hover your mouse at the top right of cell **Cmd 5**, and click on the **Add Cell Below** icon. A new cell will appear named **Cmd6**.
2. In the Notebook, in the cell **Cmd 6**, copy the following code and paste it into the cell:

   ```
   This code connects to the Data Lake Storage filesystem named "Data" and reads data in the preferen
   ```

3. In the Notebook, in the cell under **Cmd 6**, click on the **down pointing arrow** icon and click on **Move up**. Repeat until the cell appears at the top of the Notebook.
4. Leave the Azure Databricks Notebook open

   > **Note** A future lab will explore how this data can be exported to another data platform technology

   Result: After you completed this exercise, you have created an annotation within a notebook.

5.7.4 Task 4: If time permits or post course review

If you have completed this lab early, the following sections provide links to content that can help you learn more about basic and advanced transformations in Azure.

If the url are inaccessible, there is a copy of the notebooks ion the *Allfiles\Labfiles\Starter\DP-200.3\Post Course Review* folder

Basic transformations

1. Within the Workspace, using the command bar on the left, select **Workspace**, **Users**, and select **your username** (the entry with house icon).
2. In the blade that appears, select the **downwards pointing chevron next to your name**, and select **Import**.
3. On the Import Notebooks dialog, select **URL below** and paste in the following URL:

   ```
   https://github.com/MicrosoftDocs/mslearn-perform-basic-data-transformation-in-azure-databricks/blob
   ```

4. Select **Import**.
5. A folder named **05.1-Basic-ETL** after the import should appear. Select that folder.
6. The folder will contain one or more notebooks that you can use to learn basic transformations using **scala** or **python**.

Follow the instructions within the notebook, until you've completed the entire notebook. Then continue with the remaining notebooks in order:

- **01-Course-Overview-and-Setup** - This notebook gets you started with your Databricks workspace.
- **02-ETL-Process-Overview** - This notebook contains exercises to help you query, large data files and visualize your results.
- **03-Connecting-to-Azure-Blob-Storage** - You perform basic aggregation and Joins in this notebook.
- **04-Connecting-to-JDBC** - This notebook lists the steps for accessing data from various sources using Databricks.
- **05-Applying-Schemas-to-JSON** - In this notebook you learn how to query JSON & Hierarchical Data with DataFrames
- **06-Corrupt-Record-Handling** - This notebook lists the exercises that help you understand how to create ADLS and use Databricks DataFrames to query and analyze this data.
- **07-Loading-Data-and-Productionalizing** - Here you use Databricks to query and analyze data stores in Azure Data Lake Storage Gen2.
- **Parsing-Nested-Data** - This notebook is located in the Optional subfolder, and includes a sample project for you explore later on in your own time.

  [Note] You'll find corresponding notebooks within the Solutions subfolder. These contain completed cells for exercises that ask you to complete one or more challenges. Refer to these if you get stuck or simply want to see the solution.

Advanced transformations

1. Within the Workspace, using the command bar on the left, select **Workspace**, **Users**, and select **your username** (the entry with house icon).
2. In the blade that appears, select the **downwards pointing chevron next to your name**, and select **Import**.
3. On the Import Notebooks dialog, select **URL below** and paste in the following URL:

   `https://github.com/MicrosoftDocs/mslearn-perform-advanced-data-transformation-in-azure-databricks/b`
4. Select **Import**.
5. A folder named **05.2-Advanced-ETL** after the import should appear. Select that folder.
6. The folder will contain one or more notebooks that you can use to learn basic transformations using **scala** or **python**.

Follow the instructions within the notebook, until you've completed the entire notebook. Then continue with the remaining notebooks in order:

- **01-Course-Overview-and-Setup** - This notebook gets you started with your Databricks workspace.
- **02-Common-Transformations** - In this notebook you perform some common data transformation using Spark built-in functions.
- **03-User-Defined-Functions** - In this notebook you perform custom transformation using user-defined functions.
- **04-Advanced-UDFs** - In this notebook you use advanced user-defined functions to perform some complex data transformations.
- **05-Joins-and-Lookup-Tables** - In this notebook you learn how to use standard and broadcast join for tables.
- **06-Database-Writes** - This notebook contains exercises to write data to a number of target databases in parallel, storing the transformed data from your ETL job.
- **07-Table-Management** - Here you handle managed and unmanaged tables to optimize your data storage.
- **Custom-Transformations** - This notebook is located in the Optional subfolder, and includes a sample project for you to explore later on in your own time.

  [Note] You'll find corresponding notebooks within the Solutions subfolder. These contain completed cells for exercises that ask you to complete one or more challenges. Refer to these if you get stuck or simply want to see the solution. # DP 200 - Implementing a Data Platform Solution

6 Lab 4 - Building Globally Distributed Databases with Cosmos DB

Estimated Time: 60 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-200.4* folder.

6.1 Lab overview

The students will be able to describe and demonstrate the capabilities that Azure Cosmos DB can bring to an organization. They will be able to create a Cosmos DB instance and show how to upload and query data through a portal and through a .Net application. They will then be able to demonstrate how to enable global scale of the Cosmos DB database.

6.2 Lab objectives

After completing this lab, you will be able to:

1. Create an Azure Cosmos DB database built to scale
2. Insert and query data in your Azure Cosmos DB database
3. Distribute your data globally with Azure Cosmos DB

6.3 Scenario

The developers and Information Services department at AdventureWorks are aware that a new service known as Cosmos DB recently released on Azure can provided planetary scale access to data in near real-time. They want to understand the capability that the service can offer and how it can bring value to AdventureWorks, and in what circumstances.

The Information Services department want to understand how the service can be setup and how data can be uploaded. The developers would like to see an example of an application that can be used to upload data to the Cosmos. Both would like to understand how the claim of planetary scale can be met.

At the end of this lab, you will:

1. Created an Azure Cosmos DB database built to scale
2. Inserted and queried data in your Azure Cosmos DB database
3. Distributed your data globally with Azure Cosmos DB

> **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.

6.4 Exercise 1: Create an Azure Cosmos DB database built to scale

Estimated Time: 10 minutes

Individual exercise

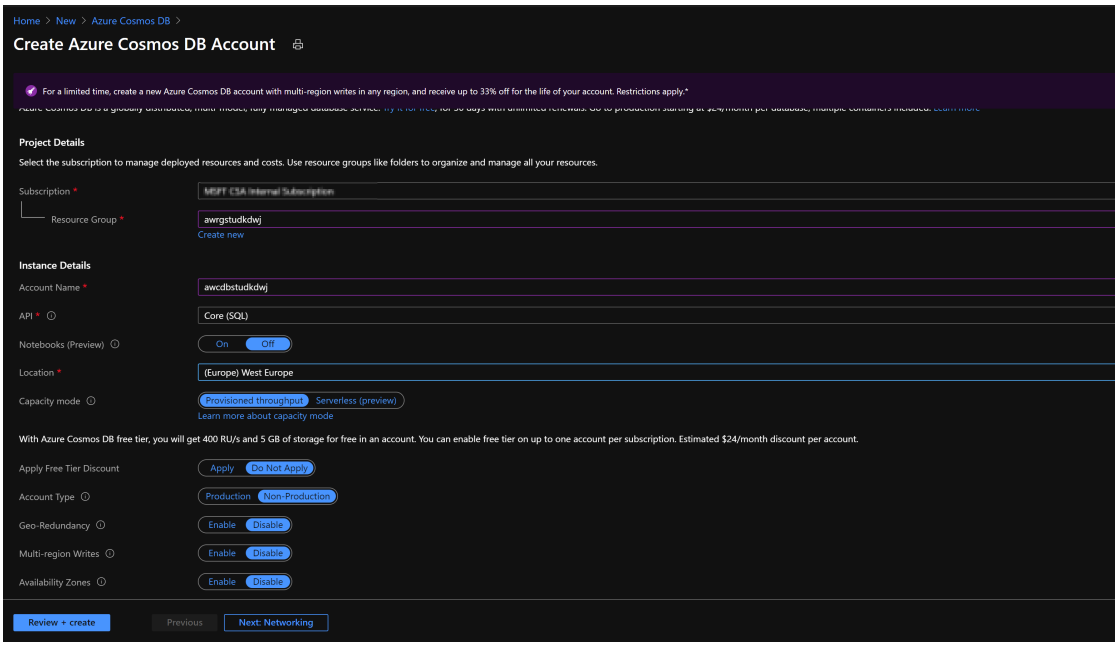
The main task for this exercise are as follows:

1. Create an Azure Cosmos DB instance

6.4.1 Task 1: Create an Azure Cosmos DB instance

1. In the Azure portal, if neccesary click on the **Home** hyperlink.
2. Navigate to the **+ Create a resource** icon.
3. In the New screen, click in the **Search the Marketplace** text box, and type the word **Cosmos**. Click **Azure Cosmos DB** in the list that appears.
4. In the **Azure Cosmos DB** screen, click **Create**.
5. From the **Create Azure Cosmos DB Account** screen, create an Azure Cosmos DB Account with the following settings:
   - In the Project details of the screen, type in the following information
     - **Subscription**: the name of the subscription you are using in this lab
     - **Resource group**: **awrgstudxx**, where **xx** are your initials
   - In the Instance details of the screen, type in the following information

- **Account name**: **awcdbstudxx**, where **xx** are your initials.
- **API**: **Core(SQL)**
- **Notebooks (Preview)**: **Off**
- **Location**: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs.
- Leave the remaining options to the default settings



6. In the **Create Azure Cosmos DB Account** blade, click **Review + create**.
7. After the validation of the **Create Azure Cosmos DB Account** blade, click **Create**.

   Note: The provision will takes approximately 5 minutes. What is often avoided in these labs is a description of the additional tabs when you provision any service in Azure. You may notice that in the provisioning screen there will be additional tabs such as Network, Tags or Advanced. This enables you to define any customized settings for a service. For example, the network tab of many services enables you to define the configuration of virtual networks, so that you are able to control and secure the network traffic against a given data service. The Tags option are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. Advanced tabs will vary dependant on the service that has it. But it is important to note that you have control over these areas and you wil want to collaborate with your Network admins or indeed your finance department to see how these options should be configured.

8. When the provisioning is complete, the "Your deployment is complete" screen appears, click on **Go to resource** and move onto the next exercise.

   Result In this exercise, you have provisioned an Azure Cosmos DB Account

6.5 Exercise 2: Insert and query data in your Azure Cosmos DB database

Estimated Time: 20 minutes

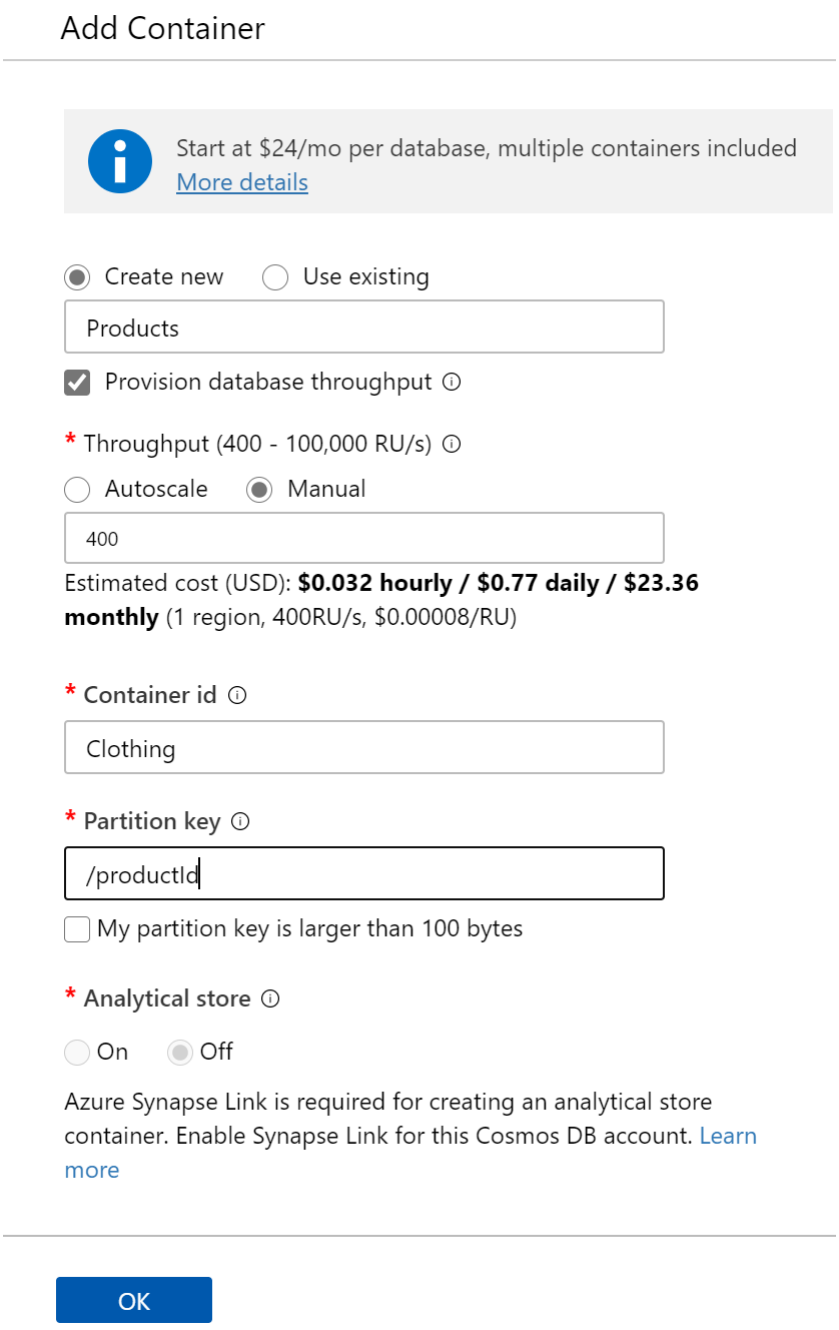
Individual exercise

The main tasks for this exercise are as follows:

1. Setup your Azure Cosmos DB database and container
2. Add data using the portal
3. Run queries in the Azure portal
4. Run complex operations on your data

6.5.1 Task 1: Setup your Azure Cosmos DB container and database

1. In the Azure portal, once the deployment of Cosmos DB is completed, click on the **Go to resources** button.
2. In the Cosmos DB screen, click on the **Overview** link.
3. In the **awcdbstudxx** screen, click **+ Add Container**. This opens up the **awcdbstudxx Data Explorer** screen with the **Add Container** blade.
4. In the **Add Container** blade, create a Products database with a container named Clothing with the following settings:
   - **Database id**: **Products**
   - **Throughput**: **400**
   - **Container id**: **Clothing**
   - **Partition key**: **/productId**
   - Leave the remaining options with their default values

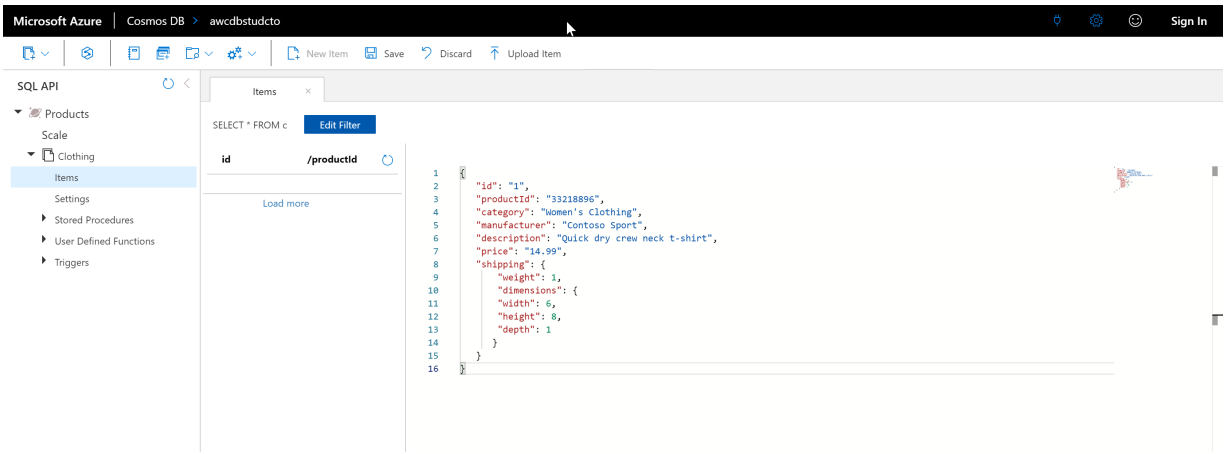


5. In the **Add Container** screen, click **OK**

6.5.2 Task 2: Add data using the portal

1. In the **awcdbstudcto - Data Explorer** screen, on the Data Explorer toolbar, opposite the button for New Container, click on the **Open Full Screen** button. In the Open Full Screen dialog box, click **Open**. A new tab opens up in Microsoft Edge.
2. In the **SQL API** pane, click in the refresh icon, and then expand **Products**, followed by **Clothing** and click on **Items**.
3. In the Documents pane, click on the icon for **New Item**. A new document appears with a sample JSON that you will now replace.
4. Copy the following code and paste it into the **Documents** tab:

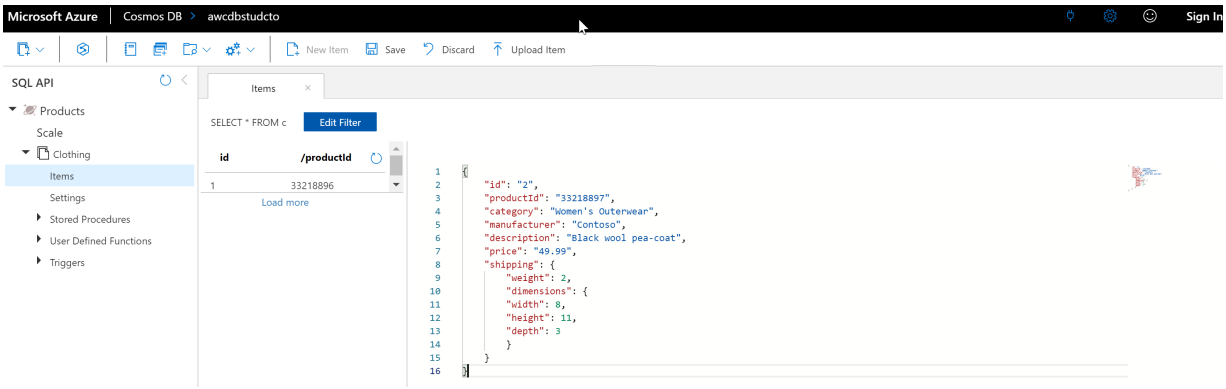
```
{
    "id": "1",
    "productId": "33218896",
    "category": "Women's Clothing",
    "manufacturer": "Contoso Sport",
    "description": "Quick dry crew neck t-shirt",
    "price": "14.99",
    "shipping": {
        "weight": 1,
        "dimensions": {
        "width": 6,
        "height": 8,
        "depth": 1
       }
    }
}
```



5. Once you've added the JSON to the Documents tab, click **Save**.
6. In the Documents pane, click on the icon for **New Item**.
7. Copy the following code and paste it into the **Items** tab:

```
{
    "id": "2",
    "productId": "33218897",
    "category": "Women's Outerwear",
    "manufacturer": "Contoso",
    "description": "Black wool pea-coat",
    "price": "49.99",
    "shipping": {
        "weight": 2,
        "dimensions": {
        "width": 8,
```

```
        "height": 11,
        "depth": 3
        }
    }
}
```



8. Once you've added the JSON to the Documents tab, click **Save**.

9. You can see each document that has been saved by clicking each document on the left-hand menu. The first item with id of 1, will have a value of **33218896**, which is named after the productid, the second item will be **33218897**

6.5.3 Task 3: Run queries in the Azure portal.

1. In the Edge browser that opened, in the data explorer, in the **Items** screen, click on the button **New SQL Query** that is above the **SQL API** Blade, above the **refresh** icon.

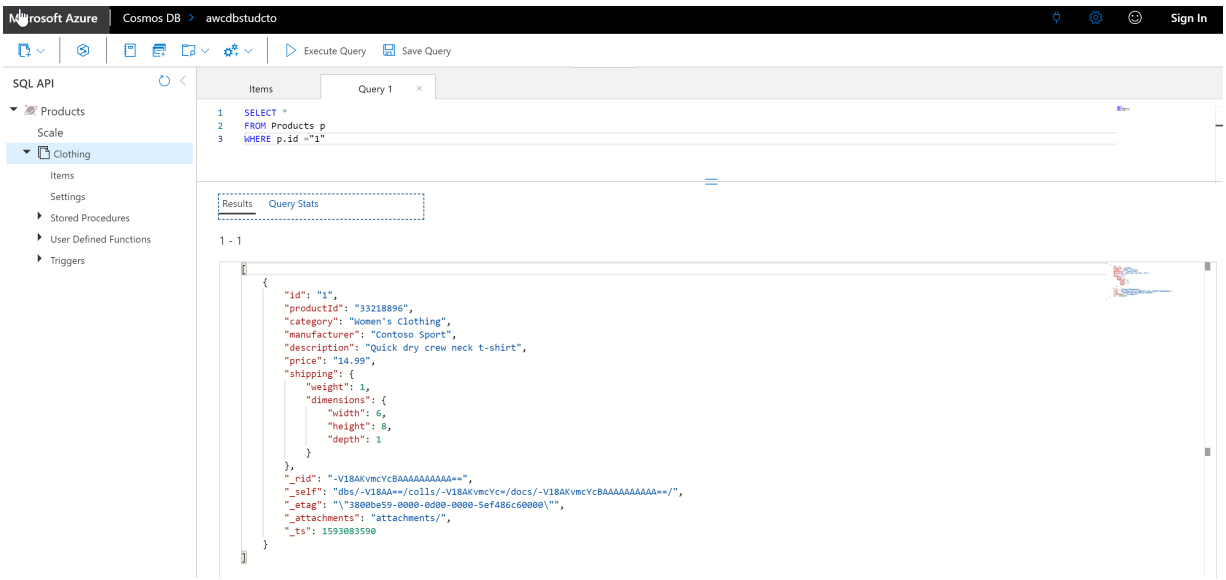
   Note: A Query 1 screen tab appears which shows the query **SELECT * FROM c** .

2. Replace the query that returns a JSON file showing details for productId 1.

```
SELECT *
FROM Products p
WHERE p.id ="1"
```

3. Click on the **Execute Query** icon. The following result is returned

```
[
    {
        "id": "1",
        "productId": "33218896",
        "category": "Women's Clothing",
        "manufacturer": "Contoso Sport",
        "description": "Quick dry crew neck t-shirt",
        "price": "14.99",
        "shipping": {
            "weight": 1,
            "dimensions": {
                "width": 6,
                "height": 8,
                "depth": 1
            }
        },
        "_rid": "I2YsALxG+-EBAAAAAAAAAA==",
        "_self": "dbs/I2YsAA==/colls/I2YsALxG+-E=/docs/I2YsALxG+-EBAAAAAAAAAA==/",
        "_etag": "\"0000844e-0000-1a00-0000-5ca79f840000\"",
        "_attachments": "attachments/",
        "_ts": 1554489220
    }
]
```

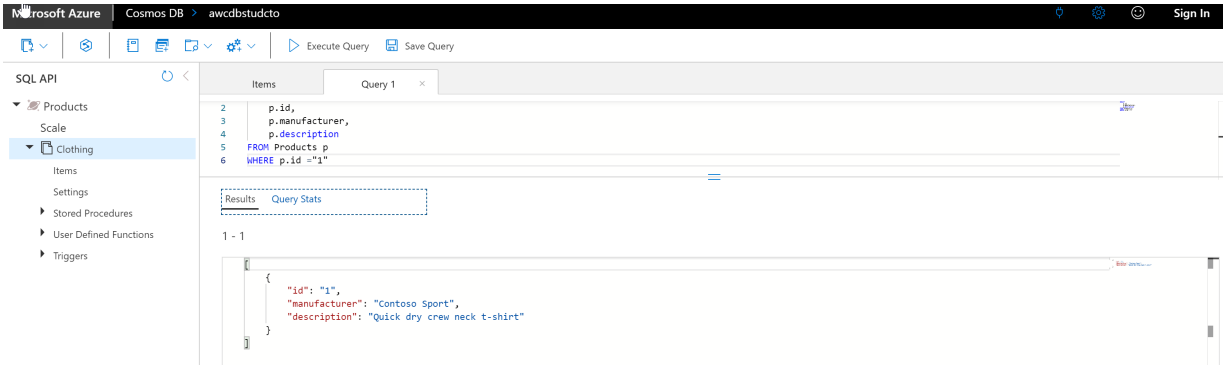


4. In the existing query window, replace the previous query and write a query that returns the id, manufacturer and description in a JSON file for productId

```
SELECT
    p.id,
    p.manufacturer,
    p.description
FROM Products p
WHERE p.id ="1"
```

5. Click on the **Execute Query** icon. The following result is returned

```
[
{
    "id": "1",
    "manufacturer": "Contoso Sport",
    "description": "Quick dry crew neck t-shirt"
}
]
```



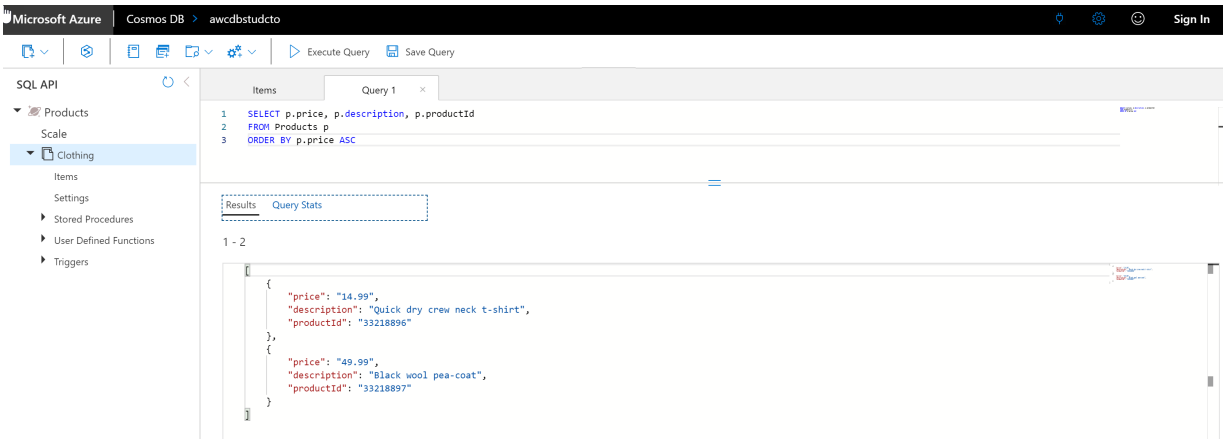
6. In the existing query window, replace the previous query and write a query that returns returns the price, description, and product ID for all products, ordered by price, in ascending order.

```
SELECT p.price, p.description, p.productId
FROM Products p
ORDER BY p.price ASC
```

7. Click on the **Execute Query** icon. The following result is returned

```
[
    {
        "price": "14.99",
        "description": "Quick dry crew neck t-shirt",
        "productId": "33218896"
```

```
    },
    {
        "price": "49.99",
        "description": "Black wool pea-coat",
        "productId": "33218897"
    }
]
```



6.5.4 Task 4: Run complex operations on your data

1. In the Edge browser that opened, in the data explorer, in the **Items** screen, click on the button **New Stored Procedure**, which you can find next to the button of open query.

   Note: A New Stored Procedure screen appears which shows a sample stored procedure .

2. In the New Stored Procedure screen, in the **Stored Procedure Id** text box, type **createMyDocument**.

3. Use the following code to create a stored procedure in the Stored Procedure Body.

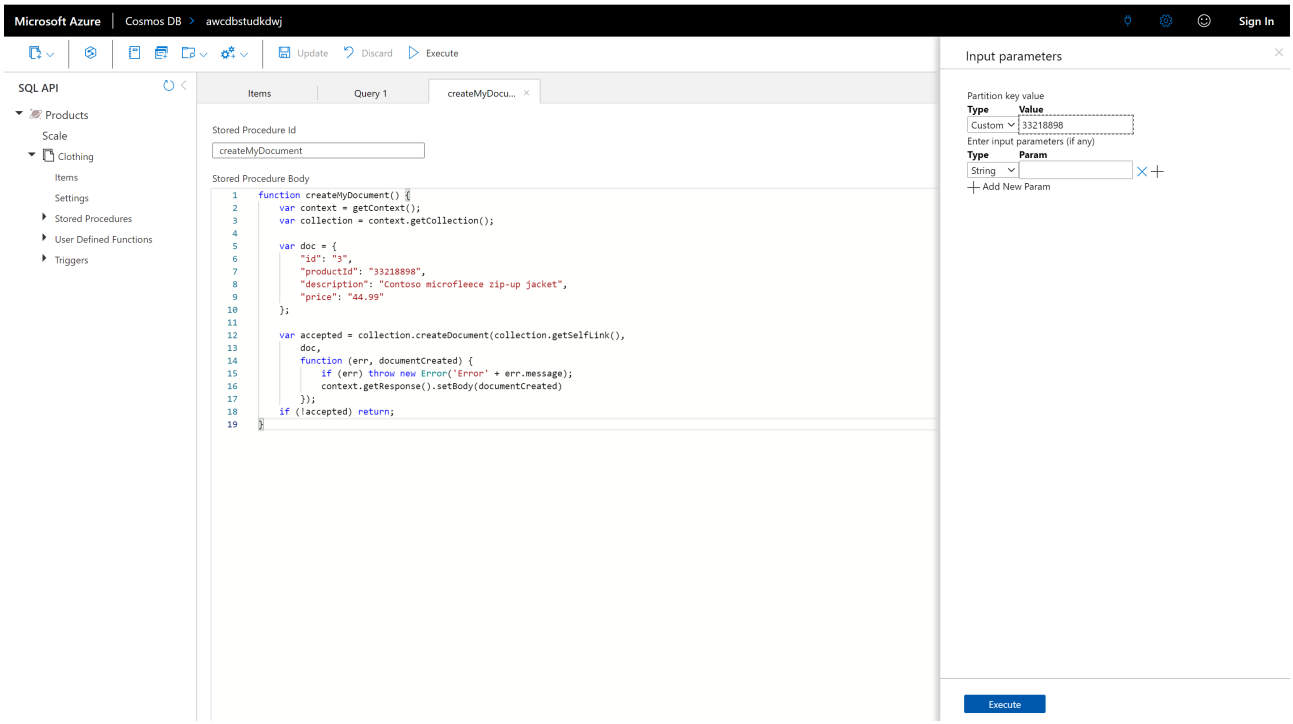
```
function createMyDocument() {
    var context = getContext();
    var collection = context.getCollection();

    var doc = {
        "id": "3",
        "productId": "33218898",
        "description": "Contoso microfleece zip-up jacket",
        "price": "44.99"
    };

    var accepted = collection.createDocument(collection.getSelfLink(),
        doc,
        function (err, documentCreated) {
            if (err) throw new Error('Error' + err.message);
            context.getResponse().setBody(documentCreated)
        });
    if (!accepted) return;
}
```

4. In the New Stored Procedure screen, click **Save**.

5. In the New Stored Procedure screen, click **Execute**.

6. In the Input Parameters screen, set the **Partition Key Value**, **Type** to **String**, and **Value** to **33218898**, leave the other settings the same, and then click **Execute**.

See below:



The following result is returned

```JSON
{
    "id": "3",
    "productId": "33218898",
    "description": "Contoso microfleece zip-up jacket",
    "price": "44.99",
    "_rid": "I2YsALxG+-EDAAAAAAAAAA==",
    "_self": "dbs/I2YsAA==/colls/I2YsALxG+-E=/docs/I2YsALxG+-EDAAAAAAAAAA==/",
    "_etag": "\"0000874e-0000-1a00-0000-5ca7a7050000\"",
    "_attachments": "attachments/"
}
```

7. In the Edge browser that opened, in the data explorer, click on the drop down button for **New Stored Procedure** and click **New UDF** .

   Note: A New UDF 1 screen appears which shows **function userDefinedFunction(){}**

8. In the New Defined Function screen, in the **User Defined Function Id** text box, type **producttax**.

9. Use the following code to create a user defined function in the user defined function Body.

```
function producttax(price) {
    if (price == undefined)
        throw 'no input';

    var amount = parseFloat(price);

    if (amount < 1000)
        return amount * 0.1;
    else if (amount < 10000)
        return amount * 0.2;
    else
        return amount * 0.4;
}
```

10. In the New UDF 1 screen, click **Save**.

11. Click on the Query 1 tab, and replace the existing query with the following query:

```
SELECT c.id, c.productId, c.price, udf.producttax(c.price) AS producttax FROM c
```

12. In the Query 1 screen, click **Execute Query**.

The following result is returned

```JSON
[
    {
        "id": "1",
        "productId": "33218896",
        "price": "14.99",
        "producttax": 1.499
    },
    {
        "id": "2",
        "productId": "33218897",
        "price": "49.99",
        "producttax": 4.9990000000000005
    },
    {
        "id": "3",
        "productId": "33218898",
        "price": "44.99",
        "producttax": 4.4990000000000005
    }
]
```

6.6 Exercise 3: Distribute your data globally with Azure Cosmos DB

Estimated Time: 15 minutes

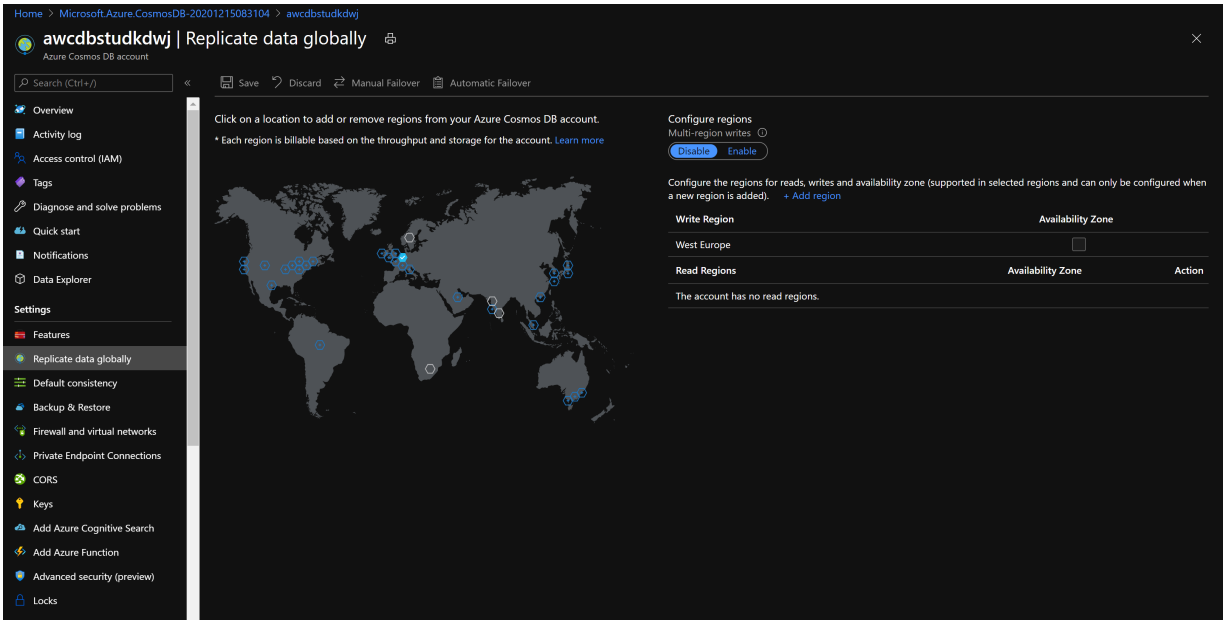
Individual exercise

The main tasks for this exercise are as follows:

1. Replicate Data to Multiple Regions
2. Managing Failover

6.6.1 Task 1: Replicate Data to Multiple Regions

1. In the Azure Portal navigate to the Cosmos DB resource **awcdbstudxx** window, in the blade **Settings**, click on **Replicate data globally**.



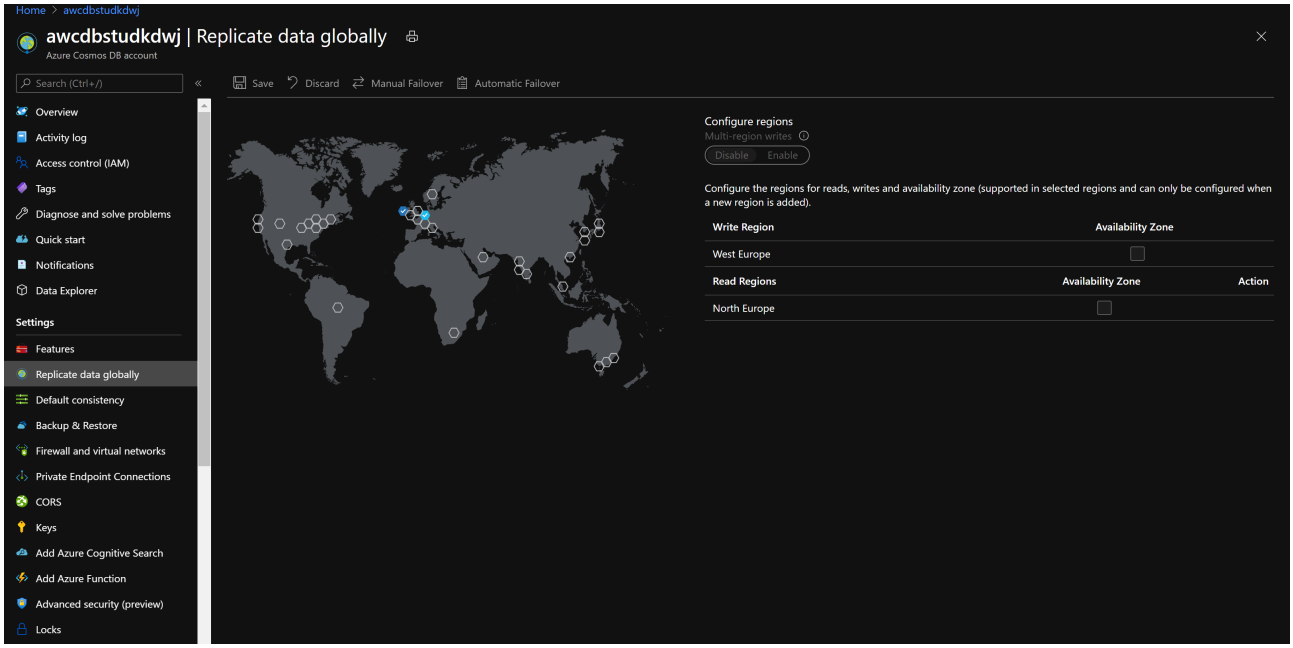
2. On the world map, single click a data center location within the continent you reside, and click on **Save**.

   Note The provisioning of the additional data centers will take approximately 7 minutes

6.6.2 Task 2: Managing Failover.

1. In the **awcdbstudxx - Replicate data globally** window, click on **Manual Failover**.
2. Click on the **Read Region** datacenter location, then click on the check box next to "I understand and agree to trigger a failover on my current Write Region.", and then click on **OK**.

   Note The Manual Failover will take approximately 3 minutes. The screen will look as follows. Note the icon colors have changed



3. In the **awcdbstudxx - Replicate data globally** window, click on **Automatic Failover**
4. In the "Automatic Failover" screen, click on the **ON** button, and then click on **OK**.

   Note The provisioning of the Automatic Failover will take approximately 3 minutes.

DP 200 - Implementing a Data Platform Solution

7 Lab 5 - Working with Relational Data Stores in the Cloud

Estimated Time: 75 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-200.5* folder.

7.1 Lab overview

The students will be able to provision an Azure SQL Database and Azure Synapse Analytics server and be able to issue queries against one of the instances that are created. They will be also be able to integrate a data warehouse with a number of other data platform technologies and use PolyBase to load data from one data source into Azure Synapse Analytics.

7.2 Lab objectives

After completing this lab, you will be able to:

1. Use Azure SQL Database
2. Describe Azure Synapse Analytics
3. Create and query Azure Synapse Analytics

4. Use PolyBase to load data into Azure Synapse Analytics

7.3 Scenario

You are the senior data engineer at AdventureWorks, and you are working with your team to transition a relational database system from an on-premises SQL Server to a Azure SQL Database located in Azure. You will begin by creating an instance of Azure SQL Database with the company's sample database. Your intention is to hand this instance of to a junior data engineer to perform some testing of departmental databases.

You will then provision Azure Synapse Analytics server and test that the provisioning of the server is successful by testing a sample database with a series of queries. You will then use PolyBase to load a dimension table from Azure Blob to test that the integration of this data platform technology with Azure Synapse Analytics.

At the end of this lad, you will have:

1. Used Azure SQL Database
2. Described Azure Synapse Analytics
3. Created and queried Azure Synapse Analytics
4. Used PolyBase to load data into Azure Synapse Analytics

   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.

7.4 Exercise 1: Use Azure SQL Database

Estimated Time: 15 minutes

Individual exercise

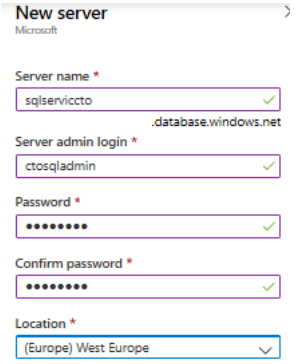
The main task for this exercise are as follows:

1. Create and configure a SQL Database instance.

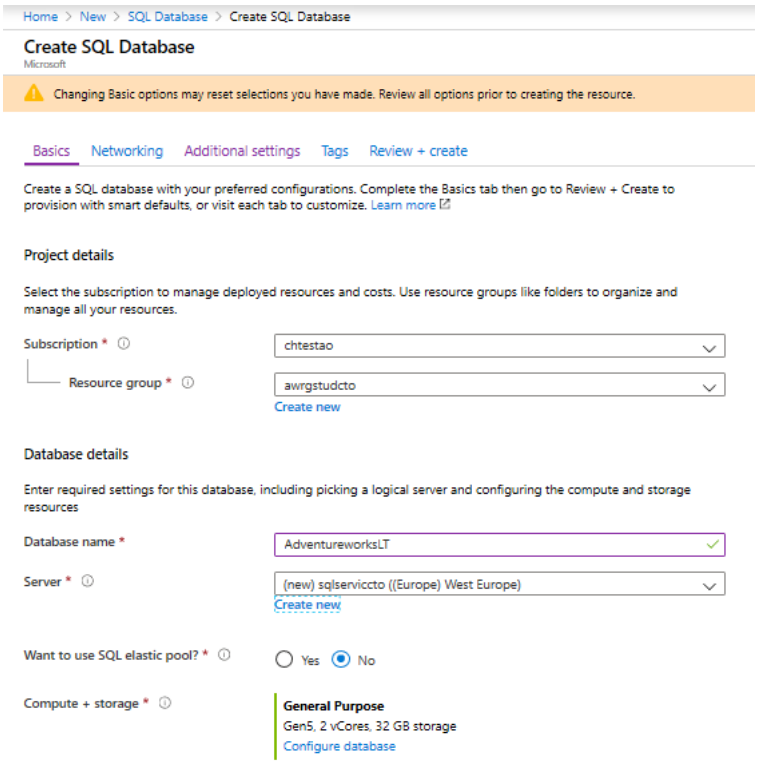
7.4.1 Task 1: Create and configure a SQL Database instance.

1. In the Azure portal, navigate to the **+ Create a resource** blade.
2. In the New screen, click the **Search the Marketplace** text box, and type the word **SQL Database**. Click **SQL Database** in the list that appears.
3. In the **SQL Database** screen, click **Create**.
4. From the **Create SQL Database** screen, create an Azure SQL Database with the following settings:
   - In the Project details section, type in the following information
     - **Subscription**: the name of the subscription you are using in this lab
     - **Resource group**: **awrgstudxx**, where **xx** are your initials.
   - Click on the **Additional setting** tab, click **Sample** . The AdventureworksLT sample database is selected automatically.
   - Click the **Basics** tab once this has been done.
   - In the Database details section, type in the following information
     - Database name: type in **AdventureworksLT**
     - Server: Create a new server by clicking **Create new** with the following settings and click on **OK**:
       * **Server name**: **sqlservicexx**, where **xx** are your initials
       * **Server admin login**: **xxsqladmin**, where **xx** are your initials
       * **Password**: **Pa55w.rd**
       * **Confirm Password**: **Pa55w.rd**
       * **Location**: choose a **location** near to you.

* click on **OK**



* Leave the remaining settings to their defaults, and then click on **OK**



5. In the **Create SQL Database** blade, click **Review + create**.
6. After the validation of the **Create SQL Database*** blade, click **Create**.

   Note: The provision will takes approximately 4 minutes.

Result: After you completed this exercise, you have an Azure SQL Database instance

7.5 Exercise 2: Describe Azure Synapse Analytics

Estimated Time: 15 minutes

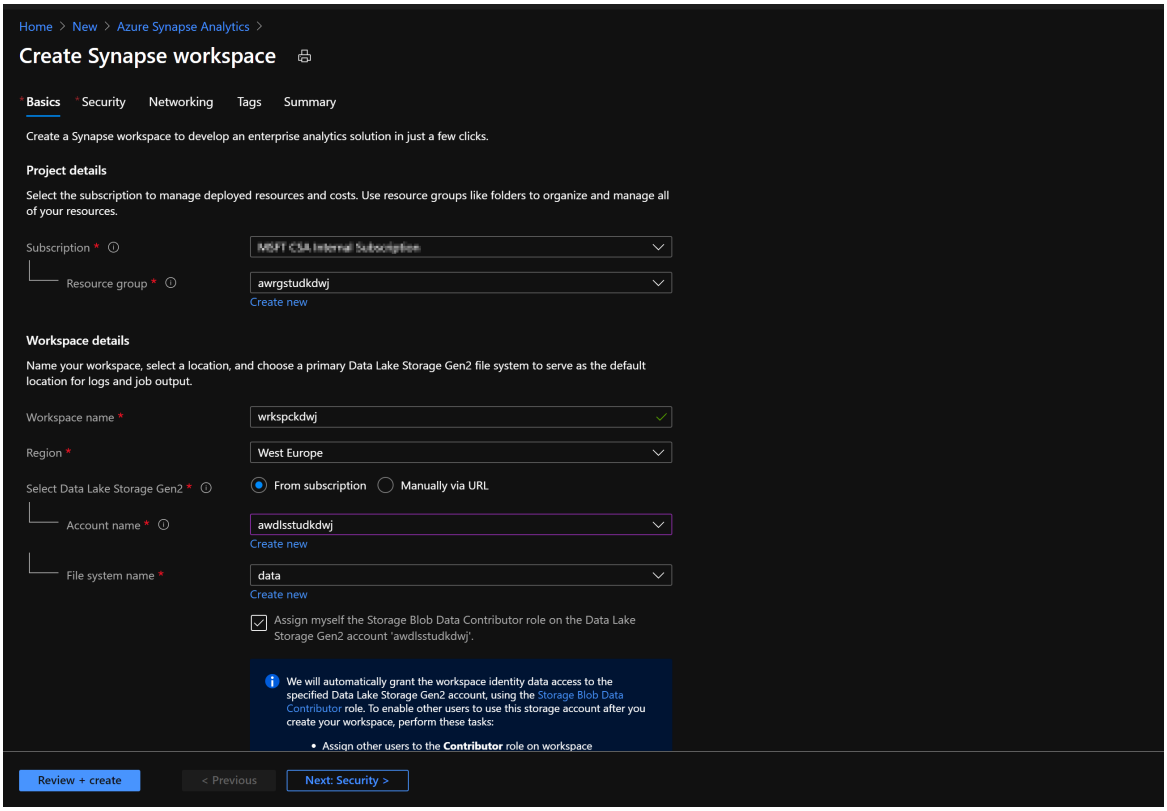
Individual exercise

The main tasks for this exercise are as follows:

1. Create and configure a Azure Synapse Analytics instance.
2. Configure the Server Firewall
3. Pause the warehouse database

7.5.1 Task 1: Create and configure a Azure Synapse Analytics instance.

1. In the Azure portal, click on the link **home** at the top left of the screen.
2. In the Azure portal, click **+ Create a resource**.
3. In the New blade, navigate to the **Search the Marketplace** text box, and type the word **Synapse**. Click **Azure Synapse Analytics** in the list that appears.
4. In the **Azure Synapse Analytics** blade, click **Create**.
5. From the **Create Synapse workspace basics** blade, create an Azure Synapse Analytics Workspace with the following settings:
   - In the Project details section, type in the following information
     - **Subscription**: the name of the subscription you are using in this lab
     - **Resource group**: **awrgstudxx**, where **xx** are your initials.
   - In the workspace details section, create the workspace with the following settings:
     - **Workspace Name**: **wrkspcxx**, where **xx** are your initials.
     - **Region**: choose the region nearest to you and where you deployed your resource group
     - **Select Data Lake Storage Gen2**: "from subscription"
     - **Account Name**: select **awdlsstudxx**, where **xx** are your initials
     - **File System Name**: select **data**
     - **Check** the "Assign myself the Storage Blob Data Contributor role on the Data Lake Storage Gen2 account 'awdlsstudxx"



   - Navigate to the **Security** tab in the **Create Synapse workspace** blade.
   - Under the SQL administrator credentials section provide the following:
     - **Password**: **Pa55w.rd**
     - **Confirm Password**: **Pa55w.rd**
     - Leave all the other settings as **default**.
   - In the screen, click **Review + create**.
   - In the blade, click **Create**.

     Note: The provision will takes approximately 7 minutes.

6. Once provisioned select **Go to resource**, and you'll be landing in the **Overview** page of your Azure Synapse Analytics workspace.
7. Select + **New dedicated SQL Pool**.

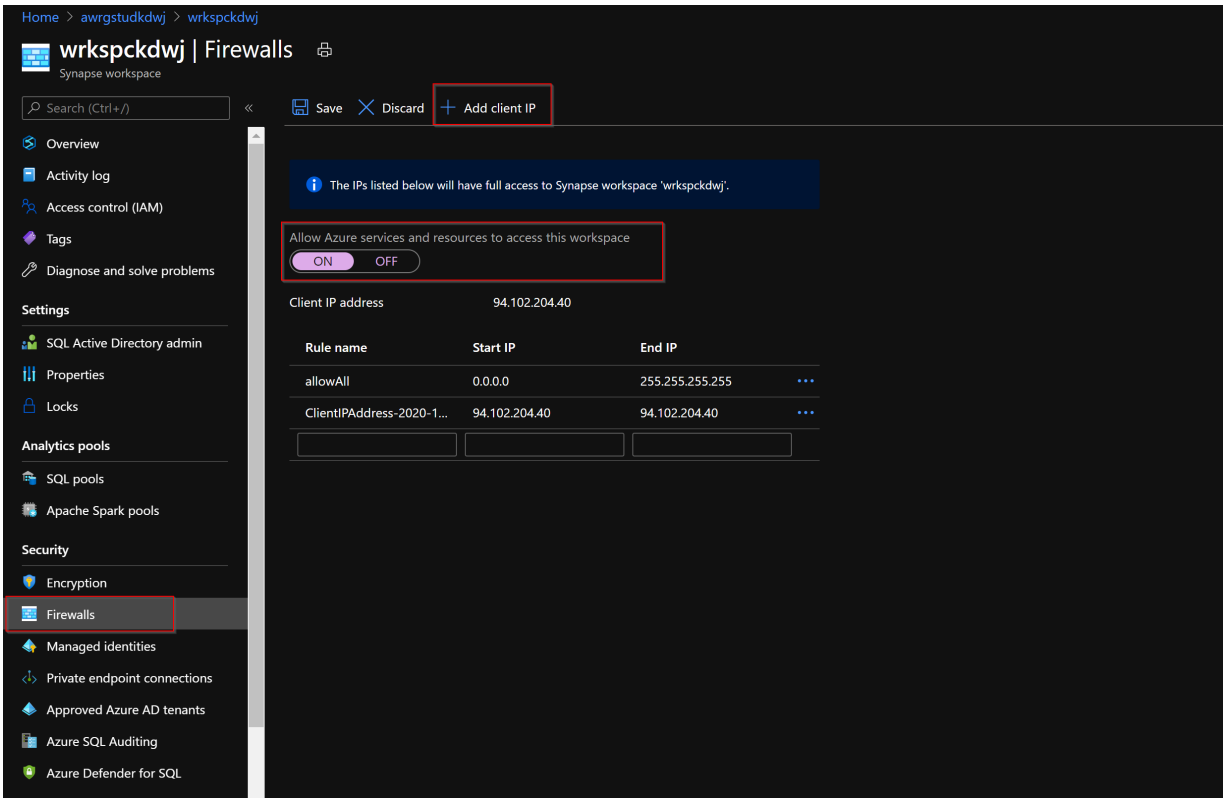
8.. In the **basics** page of **Create dedicated SQL pool** blade configure the following settings: - Dedicated SQL pool name: **dedsqlxx**, where **xx** are your initials - Leave all the other settings per default

9. In the **Create dedicated SQL pool** screen, click **Review + create**.
10. In the **Create dedicated SQL pool** blade, click **Create**.

    Note: The provision will takes approximately 7 minutes.

7.5.2 Task 2: Configure the Server Firewall

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **wrkspcxx**, where **xx** are your initials
2. In the **wrkspcxx** screen, click on **Firewalls**.
3. In the **wrkspcxx**- Firewalls screen, click on the option + **Add client IP**, and check that **Allow Azure services and resources to access this workspace** is set to **On**, and then click on **Save**. On the success screen click **OK**.



    Note: You will receive a message stating that the the server firewall rules have been successfully updated

4. Close down the Firewalls screen.

    Result: After you completed this exercise, you have created an Azure Synapse Analytics instance and configures the server firewall to enable connections against it.

7.5.3 Task 3: Pause the dedsqlxx dedicated SQL Pool

1. Navigate to **dedsqlxx** resource in your resource group.
2. Click on **dedsqlxx**, where **xx** are your initials.
3. In the **dedsqlkxx (wrkspcxx/dedsqlxx)** screen, click on **Pause**.
4. In the Pause **dedsqlxx** screen, click **Yes**

7.6 Exercise 3: Creating an Azure Synapse Analytics database and tables

Estimated Time: 25 minutes

Individual exercise

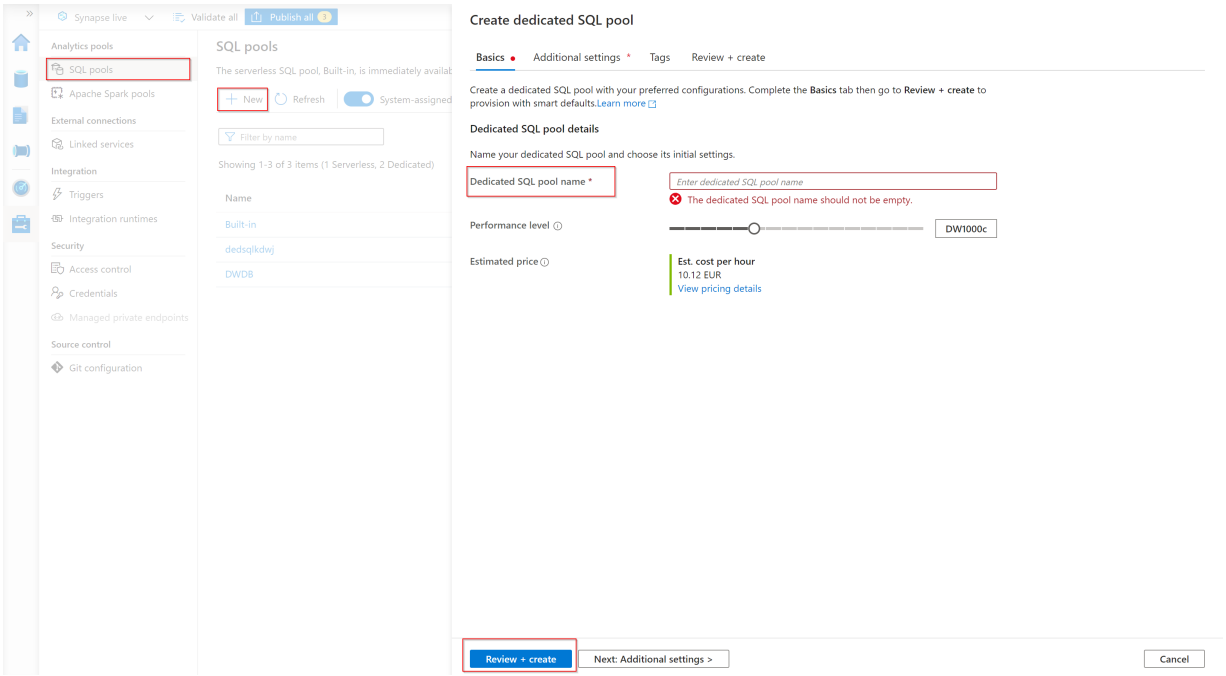
The main tasks for this exercise are as follows:

1. Understand Synapse Studio and connect to a dedicated SQL Pool.
2. Create a dedicated SQL Pool database
3. Create dedicated SQL Pool tables

> **Note**: If you are not familiar with Transact-SQL, statements are available for the following labs in the following location **Allfiles\Labfiles\Starter\DP-200.5\SQL DW Files**

7.6.1 Task 1: Connect the Dedicated SQL Pool to Azure Synapse Studio

1. Navigate to the **dedsqlxx** resource in your resource group.
2. In the **overview** section of the Synapse Workspace navigate to **Launch Synapse Studio**
3. Click on the **Manage Hub** on the left side of the screen
4. Select **SQL pool** and select **+ New**
   - In the Basics details section, type in the following information
     - **Dedicated SQL pool name**: **DWDB**
   - Leave all the other settings as default
   - Select **Review + Create** and select **Create**



> **Note**: The creation of the database takes approximately 6 minutes.

5. Once the database is set up, navigate to the **Data Hub** on the left side of the screen. Select the ellipsis next to **Databases** and select **refresh**. You should see the the newly created database DWDB.

7.6.2 Task 3: Create dedicated SQL Pool tables.

1. In Synapse Studio, navigate to the newly created database under **Databases** in the **Data hub**, when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
2. Select the ellipsis next to the **DWDB** database.
3. Select **New SQL script**, and **empty script**

Note: If you are unfamiliar with Transact-SQL, there is a script in the Allfiles\Solution\DP-200.5\folder named **Exercise3 Task3Step2 script.sql**. It contains the bulk of the code required to create the tables, but you do have to complete the code by selecting the distribution type to use for each table

4. Create a table named **dbo.Users** with a **clustered columnstore** index with a distribution of **replicate** with the following columns:

column name	data type	Nullability
userId	int	NULL
City	nvarchar(100)	NULL
Region	nvarchar(100)	NULL
Country	nvarchar(100)	NULL

Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

5. In **Synapse Studio**, click on **Run** and the query will be executed. To verify if the **dbo.Users** table was created you can click refresh and navigate to **tables** which, when expanded, should show you the table.
6. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
7. Select the ellipsis next to the **DWDB** database.
8. Select **New SQL script**, and **empty script**
9. Create a table named **dbo.Products** with a **clustered columnstore** index with a distribution of **round robin** with the following columns:

column name	data type	Nullability
ProductId	int	NULL
EnglishProductName	nvarchar(100)	NULL
Color	nvarchar(100)	NULL
StandardCost	int	NULL
ListPrice	int	NULL
Size	nvarchar(100)	NULL
Weight	int	NULL
DaysToManufacture	int	NULL
Class	nvarchar(100)	NULL
Style	nvarchar(100)	NULL

Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

10. In **Synapse Studio**, click on **Run** and the query will be executed. To verify if the **dbo.Products** table was created you can click refresh and navigate to **tables** which, when expanded, should show you the table.
11. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
12. Select the ellipsis next to the **DWDB** database.
13. Select **New SQL script**, and **empty script**
14. Create a table named **dbo.FactSales** with a **clustered columnstore** index with a distribution of **Hash** on the **SalesUnit** with the following columns:

column name	data type	Nullability
DateId	int	NULL
ProductId	int	NULL
UserId	int	NULL
UserPreferenceId	int	NULL
SalesUnit	int	NULL

Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

15. In **Synapse Studio**, click on **Run** and the query will be executed. To verify if the **dbo.FactSales** table was created you can click refresh and navigate to **tables** which, when expanded, should show you the table.

    Result: After you completed this exercise, you have used Synapse Studio to create a data warehouse named DWDB and three tables named Users, Products and FactSales.

7.7 Exercise 4: Using PolyBase to Load Data into Azure Synapse Analytics

Estimated Time: 10 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Collect Data Lake Storage container and key details
2. Create a dbo.Dates table using PolyBase from Azure Data Lake Storage

7.7.1 Task 1: Collect Azure Blob account name and key details

1. In the Azure portal, click on **Resource groups** and then click on **awrgstudxx**, and then click on **awdlsstudxx** where xx are the initials of your name.
2. In the **awdlsstudxx** screen, click **Access keys**. Click on the icon next to the **Storage account name** and paste it into Notepad.
3. In the **awdlsstudxx - Access keys** screen, under **key1**, Click on the icon next to the **Key** and paste it into Notepad.

7.7.2 Task 2: Create a dbo.Dates table using PolyBase from Azure Blob

1. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
2. Select the ellipsis next to the **DWDB** database.
3. Select **New SQL script**, and **empty script**
4. Create a **master key** against the **DWDB** database. In the query editor, type in the following code:

```
CREATE MASTER KEY;
```

    Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

5. In **Synapse Studio**, click on **Run** and the query will be executed.
6. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
7. Select the ellipsis next to the **DWDB** database.
8. Select **New SQL script**, and **empty script**
9. Create a database scoped credential named **AzureStorageCredential** with the following details, by typing in the following code in the query editor:
    - IDENTITY: **MOCID**
    - SECRET: **The access key of your storage account**

```
CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential
WITH
IDENTITY = 'MOCID',
SECRET = 'Your storage account key'
```

; “‘ >**Note**: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

10. In **Synapse Studio**, click on **Run** and the query will be executed.
11. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.

12. Select the ellipsis next to the **DWDB** database.
13. Select **New SQL script**, and **empty script**
14. In the Query window, type in code that will create an external data source named **AzureStorage** for the Blob storage account and data container created in with a type of **HADOOP** that makes use of the **AzureStorageCredential**. Note that you should replace **awdlsstudxx** in the location key with your storage account with your initials

```
CREATE EXTERNAL DATA SOURCE AzureStorage
WITH (
    TYPE = HADOOP,
    LOCATION = 'abfs://data@awdlsstudxx.dfs.core.windows.net',
    CREDENTIAL = AzureStorageCredential
);
```

    Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

15. In **Synapse Studio**, click on **Run** and the query will be executed.
16. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
17. Select the ellipsis next to the **DWDB** database.
18. Select **New SQL script**, and **empty script**
19. In the Query window, type in code that will create an external file format named **TextFile** with a formattype of **DelimitedText** and a filed terminator of **comma**.

```
CREATE EXTERNAL FILE FORMAT TextFile
WITH (
    FORMAT_TYPE = DelimitedText,
    FORMAT_OPTIONS (FIELD_TERMINATOR = ',')
);
```

    Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

20. In **Synapse Studio**, click on **Run** and the query will be executed.
21. . In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
22. Select the ellipsis next to the **DWDB** database.
23. Select **New SQL script**, and **empty script**
24. In the Query window, type in code that will create an external table named **dbo.DimDate2External** with the **location** as the root file, the Data source as **AzureStorage**, the File_format of **TextFile** with the following columns:

column name	data type	Nullability
Date	datetime2(3)	NULL
DateKey	decimal(38, 0)	NULL
MonthKey	decimal(38, 0)	NULL
Month	nvarchar(100)	NULL
Quarter	nvarchar(100)	NULL
Year	decimal(38, 0)	NULL
Year-Quarter	nvarchar(100)	NULL
Year-Month	nvarchar(100)	NULL
Year-MonthKey	nvarchar(100)	NULL
WeekDayKey	decimal(38, 0)	NULL
WeekDay	nvarchar(100)	NULL
Day Of Month	decimal(38, 0)	NULL

```
CREATE EXTERNAL TABLE dbo.DimDate2External (
[Date] datetime2(3) NULL,
[DateKey] decimal(38, 0) NULL,
```

```
[MonthKey] decimal(38, 0) NULL,
[Month] nvarchar(100) NULL,
[Quarter] nvarchar(100) NULL,
[Year] decimal(38, 0) NULL,
[Year-Quarter] nvarchar(100) NULL,
[Year-Month] nvarchar(100) NULL,
[Year-MonthKey] nvarchar(100) NULL,
[WeekDayKey] decimal(38, 0) NULL,
[WeekDay] nvarchar(100) NULL,
[Day Of Month] decimal(38, 0) NULL
)
WITH (
    LOCATION='/DimDate2.txt',
    DATA_SOURCE=AzureStorage,
    FILE_FORMAT=TextFile
);
```

Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

25. In **Synapse Studio**, click on **Run** and the query will be executed.
26. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
27. Select the ellipsis next to the **DWDB** database.
28. Select **New SQL script**, and **empty script**
29. Test that the table is created by running a select statement against it:

    ```
    SELECT * FROM dbo.DimDate2External;
    ```

    Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.
30. In **Synapse Studio**, click on **Run** and the query will be executed.
31. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
32. Select the ellipsis next to the **DWDB** database.
33. Select **New SQL script**, and **empty script**
34. In the Query window, type in a **CTAS** statement that creates a table named **dbo.Dates** with a **column-store** index and a **distribution** of **round robin** that loads data from the **dbo.DimDate2External** table.

    ```
    CREATE TABLE dbo.Dates
    WITH
    (
        CLUSTERED COLUMNSTORE INDEX,
        DISTRIBUTION = ROUND_ROBIN
    )
    AS
    SELECT * FROM [dbo].[DimDate2External];
    ```

    Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.
35. In **Synapse Studio**, click on **Run** and the query will be executed.
36. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
37. Select the ellipsis next to the **DWDB** database.
38. Select **New SQL script**, and **empty script**
39. In the Query window, type in a query that creates statistics on the **DateKey**, **Quarter** and **Month** column.

```
CREATE STATISTICS [DateKey] on [Dates] ([DateKey]);
CREATE STATISTICS [Quarter] on [Dates] ([Quarter]);
CREATE STATISTICS [Month] on [Dates] ([Month]);
```

Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

40. In **Synapse Studio**, click on **Run** and the query will be executed.
41. In Synapse Studio, navigate to the newly created database under **Databases** , when opening the ellipsis in the **Data hub** tab. Click on **DWDB**.
42. Select the ellipsis next to the **DWDB** database.
43. Select **New SQL script**, and **empty script**
44. Test that the table is created by running a select statement against it

```
SELECT * FROM dbo.Dates;
```

Note: Make sure that the script has is connected to **DWDB** and uses the database **DWDB**.

45. In **Synapse Studio**, click on **Run** and the query will be executed.

DP 200 - Implementing a Data Platform Solution

8 Lab 6 - Performing Real-Time Analytics with Stream Analytics

Estimated Time: 60 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-200.6* folder.

8.1 Lab overview

The students will be able to describe what data streams are and how event processing works and choose an appropriate data stream ingestion technology for the AdventureWorks case study. They will provision the chosen ingestion technology and integrate this with Stream Analytics to create a solution that works with streaming data.

8.2 Lab objectives

After completing this lab, you will be able to:

1. Explain data streams and event processing
2. Ingest data with Event Hubs
3. Initiate a data generation application
4. Process Data with a Stream Analytics Jobs

8.3 Scenario

As part of the digital transformation project, you have been tasked by the CIO to help the customer services departments identify fraudulent calls. Over the last few years the customer services departments have observed an increase in calls from fraudulent customer who are asking for support for bikes that are no longer in warranty, or bikes that have not even been purchased at AdventureWorks.

The department are currently relying on the experience of customer services agents to identify this. As a result, they would like to implement a system that can help the agents track in real-time who could be making a fradulent claim.

At the end of this lad, you will have:

1. Explained data streams and event processing
2. Ingested data with Event Hubs
3. Initiated a data generation application
4. Processed Data with Stream Analytics Jobs

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.

8.4 Exercise 1: Explain data streams and event processing

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.

8.4.1 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.

8.4.2 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 .

8.5 Exercise 2: Data Ingestion with Event Hubs.

Estimated Time: 15 minutes

Individual exercise

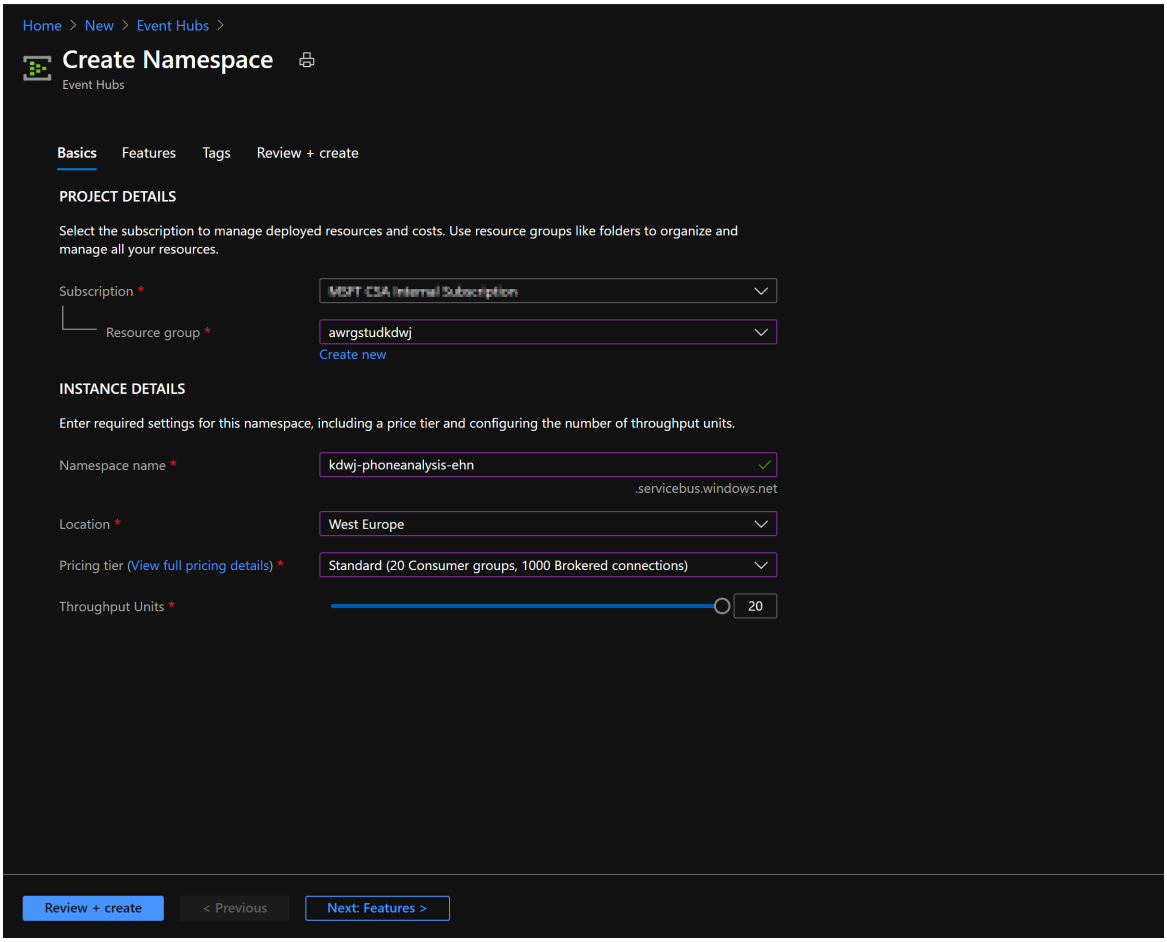
The main tasks for this exercise are as follows:

1. Create and configure an Event Hub Namespace.
2. Create and configure an Event Hub.
3. Configure Event Hub security.

8.5.1 Task 1: Create and configure an Event Hub Namespace.

1. In the Azure portal, click on the **Home** hyperlink at the top left of the screen.
2. In the Azure portal, click on the **+ Create a resource** icon , type **Event Hubs**, and then select **Event Hubs** from the resulting search. In the Event Hubs screen, click **Create**.
3. In the Create Namespace blade, type out the following options:
   - **Subscription**: **Your subscription**
   - **Resource group**: **awrgstudxx**
   - **Namespace Name**: **xx-phoneanalysis-ehn**, where xx are your initials
   - **Location**: select the location closest to you
   - **Pricing Tier**: **Standard**
   - **Throughput Units**: **20**

- Leave other options to their default settings

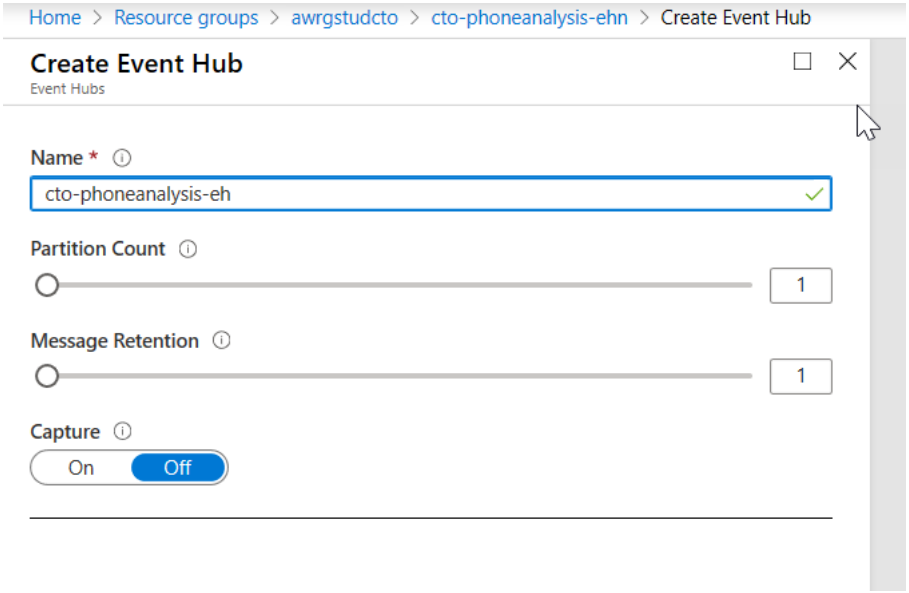


4. Then click **Review + Create** and then click **Create**

   Note: The creation of the Event Hub Namespace takes approximately 1 minute.

8.5.2 Task 2: Create and configure an Event Hub

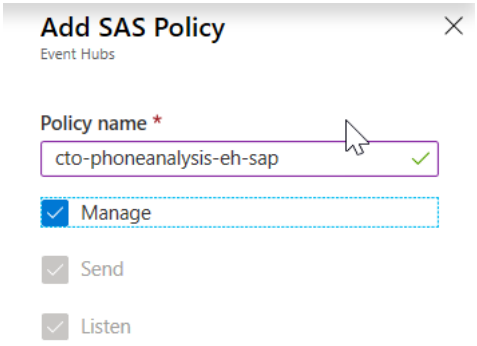
1. In the Azure portal, click on the **Home** hyperlink at the top left of the screen.
2. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, where **xx** are your initials
3. Click on **xx-phoneanalysis-ehn**, where **xx** are your initials.
4. In the **xx-phoneanalysis-ehn** screen, click on **+ Event Hubs**.
5. Provide the name **xx-phoneanalysis-eh**, leave the other settings to thier default values and then select **Create**.



Note: You will receive a message stating that the Event Hub is created after about 10 seconds

8.5.3 Task 3: Configure Event Hub security

1. In the Azure portal, in the **xx-phoneanalysis-ehn** screen, where **xx** are your initials. Scroll to the bottom of the window, and click on **xx-phoneanalysis-eh** event hub.
2. To grant access to the event hub, in the blade under the section **settings** on the left click **Shared access policies**.
3. Under the **xx-phoneanalysis-eh - Shared access policies** screen, create a policy with **Manage** permissions by selecting + **Add**. Give the policy the name of **xx-phoneanalysis-eh-sap** , check **Manage**, and then click **Create**.



4. Click on your new policy **xx-phoneanalysis-eh-sap** after it has been created, and then select the copy button for the **CONNECTION STRING - PRIMARY KEY** and paste the CONNECTION STRING - PRIMARY KEY into Notepad, this is needed later in the exercise.

   NOTE: The connection string looks as follows:

   ```
   Endpoint=sb://<Your event hub namespace>.servicebus.windows.net/;SharedAccessKeyName=<Your sha
   ```

   Notice that the connection string contains multiple key-value pairs separated with semicolons: Endpoint, SharedAccessKeyName, SharedAccessKey, and EntityPath.

5. Close down the Event hub screens in the portal

Result: After you completed this exercise, you have created an Azure Event Hub within an Event Hub Namespace and set the security for the Event Hub that can be used to provide access to the service.

8.6 Exercise 3: Starting the telecom event generator application

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Updates the application connection string
2. Run the application

8.6.1 Task 1: Updates the application connection string.

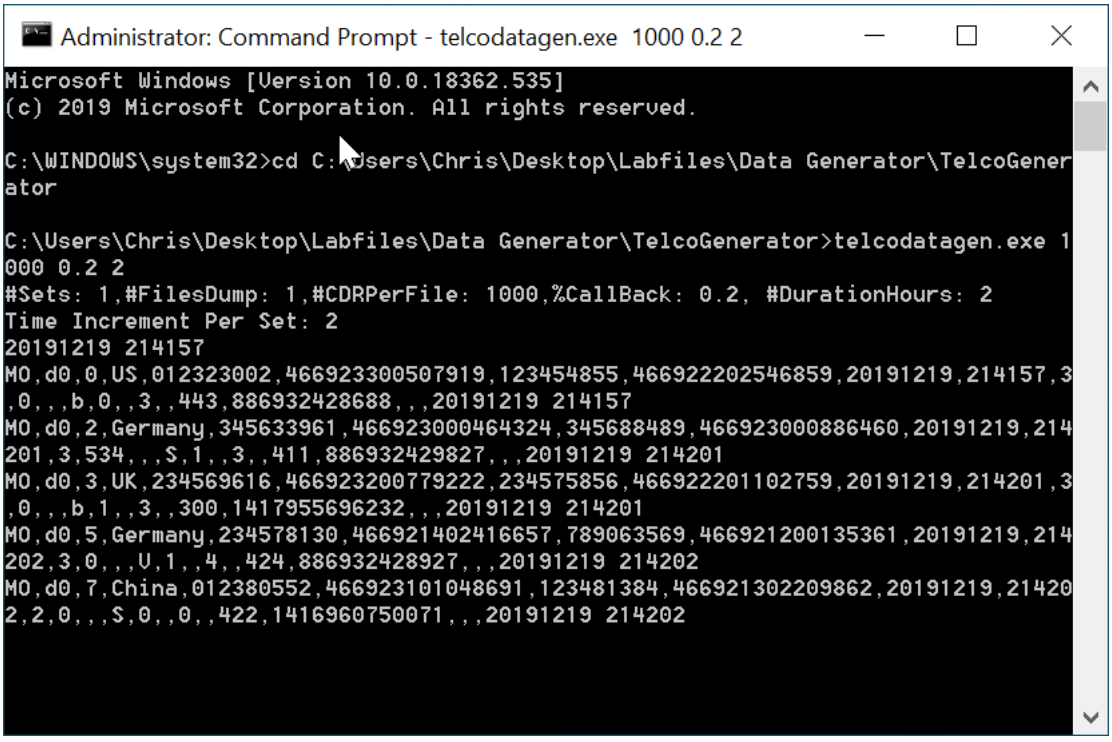
1. Browse to the location **\Labfiles\Starter\DP-200.6\DataGenerator**
2. Open the **telcodatagen.exe.config** file in a text editor of your choice
3. Update the element in the config file with the following details:
   - Set the value of the **EventHubName** key to the value of the **EntityPath** in the connection string.
   - Set the value of the **Microsoft.ServiceBus.ConnectionString** key to the connection string **without the EntityPath value** (don't forget to remove the semicolon that precedes it).
4. Save the file.

8.6.2 Task 2: Run the application.

1. Click on **Start**, and type **CMD**
2. Right click **Command Prompt**, click **Run as Administer**, and in the User Access Control screen, click **Yes**
3. In Command Prompt, browse to the location **\Labfiles\Starter\DP-200.6\DataGenerator**
4. Type in the following command:

   ```
   telcodatagen.exe 1000 0.2 2
   ```

   NOTE: This command takes the following parameters: Number of call data records per hour. Percentage of fraud probability, which is how often the app should simulate a fraudulent call. The value 0.2 means that about 20% of the call records will look fraudulent. Duration in hours, which is the number of hours that the app should run. You can also stop the app at any time by ending the process (Ctrl+C) at the command line.



After a few seconds, the app starts displaying phone call records on the screen as it sends them to the event hub. The phone call data contains the following fields:

Record	Definition
CallrecTime	The timestamp for the call start time.
SwitchNum	The telephone switch used to connect the call. For this example, the switches are strings that represent the
CallingNum	The phone number of the caller.
CallingIMSI	The International Mobile Subscriber Identity (IMSI). It's a unique identifier of the caller.
CalledNum	The phone number of the call recipient.
CalledIMSI	International Mobile Subscriber Identity (IMSI). It's a unique identifier of the call recipient.

5. Minimize the command prompt window.

   Result: After you completed this exercise, you have conmfigured an application to generate data to minimic phone calls recieved by a call center.

8.7 Exercise 4: Processing Data with Stream Analytics Jobs

Estimated Time: 15 minutes

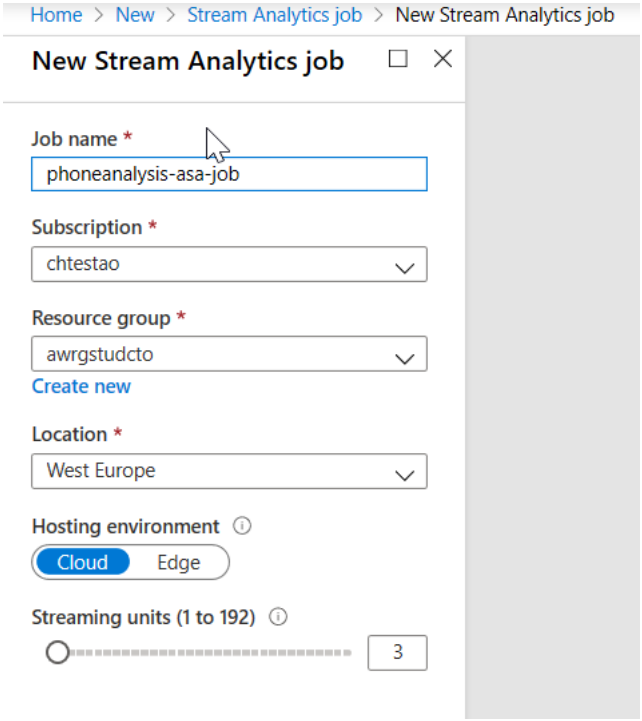
Individual exercise

The main tasks for this exercise are as follows:

1. Provision a Stream Analytics job.
2. Specify the a Stream Analytics job input.
3. Specify the a Stream Analytics job output.
4. Defining a Stream Analytics query.
5. Start the Stream Analytics job.
6. Validate streaming data is collected

8.7.1 Task 1: Provision a Stream Analytics job.

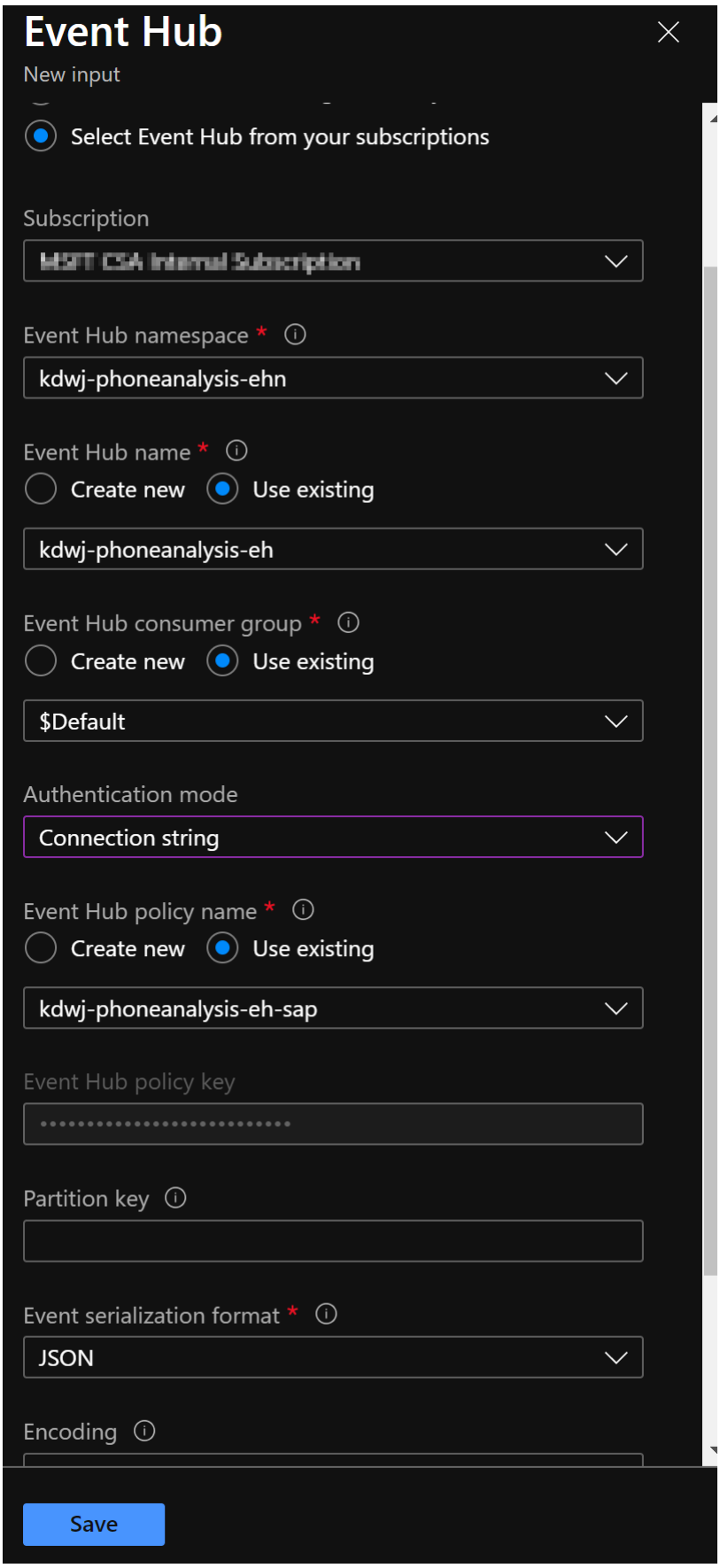
1. Go back to the Azure portal, navigate and click on the **+ Create a resource** icon, type **STREAM**, and then click the **Stream Analytics Job**, and then click **Create**.
2. In the **New Stream Analytics job** screen, fill out the following details and then click on **Create**:
   - **Job name**: phoneanalysis-asa-job.
   - **Subscription**: select your subscription
   - **Resource group**: awrgstudxx
   - **Location**: choose a location nearest to you.
   - Leave other options to their default settings



Note: You will receive a message stating that the Stream Analytics job is created after about 10 seconds. It may take a couple of minutes to update in the Azure portal.

8.7.2 Task 2: Specify the a Stream Analytics job input.

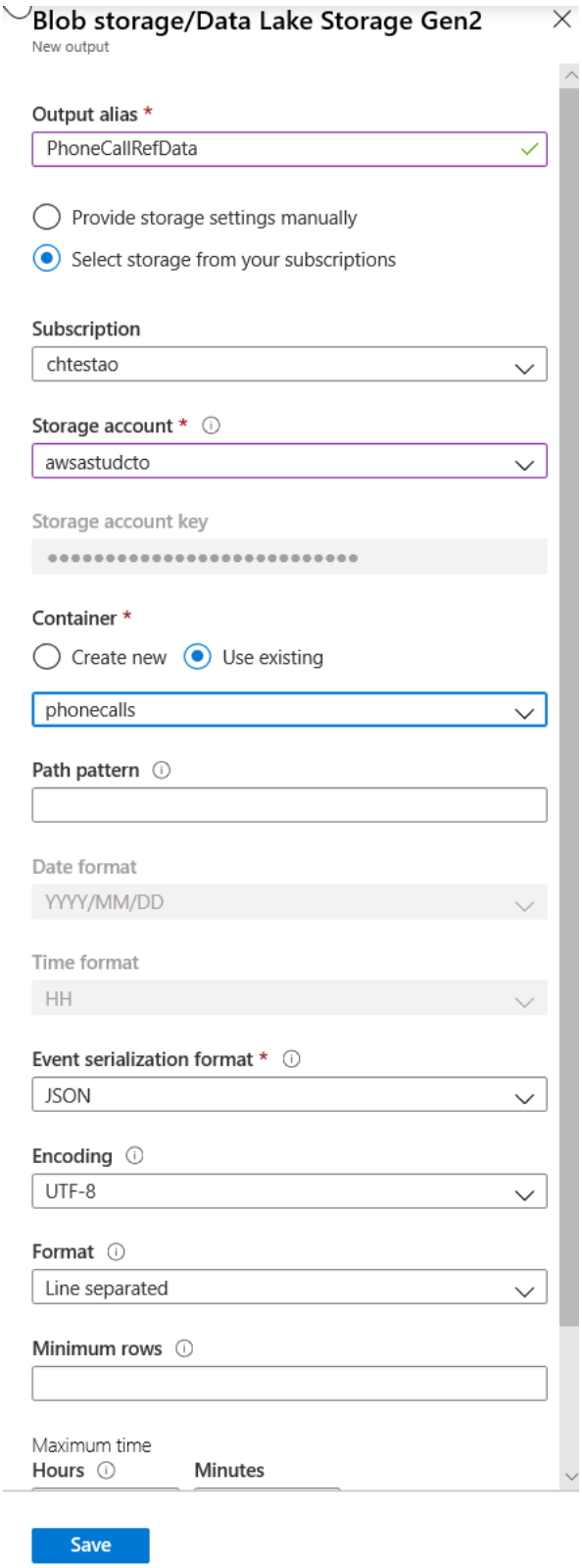
1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, where **xx** are your initials.
2. Click on **phoneanalysis-asa-job**.
3. In your **phoneanalysis-asa-job** Stream Analytics job window, in the left hand blade, under **Job topology**, click **Inputs**.
4. In the **Inputs** screen, click **+ Add stream input**, and then click **Event Hubs**.
5. In the Event Hub screen, type in the following values and click the **Save** button.
   - **Input alias**: Enter a name for this job input as **PhoneStream**.
   - **Select Event Hub from your subscriptions**: checked
   - **Subscription**: Your subscription name
   - **Event Hub Namespace**: xx-phoneanalysis-ehn
   - **Event Hub Name**: Use existing named xx-phoneanalysis-eh
   - **Event Hub Consumer Group**: Use existing
   - **Authentication Method**: Connection string
   - **Event Hub Policy Name**: use existing named xx-phoneanalysis-eh-sap
   - Leave the rest of the entries as default values. Finally, click **Save***.



6. Once completed, the **PhoneStream** Input job will appear under the input window. Close the input widow to return to the Resource Group Page

8.7.3 Task 3: Specify the a Stream Analytics job output.

1. Click on **phoneanalysis-asa-job**.
2. In your **phoneanalysis-asa-job** Stream Analytics job window, in the left hand blade, under **Job topology**, click **Outputs**.
3. In the **Outputs** screen, click **+ Add**, and then click **Blob storage/ADLS Gen2**.
4. In the **Blob storage/ADLS Gen2** window, type or select the following values in the pane:
   - **Output alias**: **PhoneCallRefData**
   - **Select Event Hub from your subscriptions**: checked
   - **Subscription**: Your subscription name
   - **Storage account**: **:awsastudxx**:, where xx is your initials
   - **Container**: **Use existing** and select **phonecalls**
   - Leave the rest of the entries as default values. Finally, click **Save**.



5. Close the output screen to return to the Resource Group page

8.7.4 Task 4: Defining a Stream Analytics query.

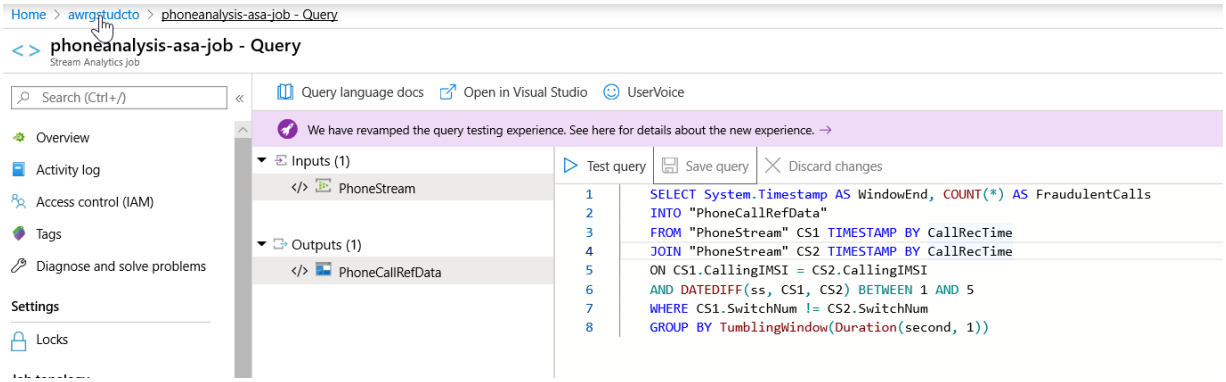
1. Click on **phoneanalysis-asa-job**.
2. In your **phoneanalysis-asa-job** window, in the **Query** screen in the middle of the window, click on **Edit query**
3. Replace the following query in the code editor:

```
SELECT
    *
INTO
    [YourOutputAlias]
FROM
    [YourInputAlias]
```

4. Replace with

```
SELECT System.Timestamp AS WindowEnd, COUNT(*) AS FraudulentCalls
INTO "PhoneCallRefData"
FROM "PhoneStream" CS1 TIMESTAMP BY CallRecTime
JOIN "PhoneStream" CS2 TIMESTAMP BY CallRecTime
ON CS1.CallingIMSI = CS2.CallingIMSI
AND DATEDIFF(ss, CS1, CS2) BETWEEN 1 AND 5
WHERE CS1.SwitchNum != CS2.SwitchNum
GROUP BY TumblingWindow(Duration(second, 1))
```

> NOTE: This query performs a self-join on a 5-second interval of call data. To check for fraudulent calls, you can self-join the streaming data based on the CallRecTime value. You can then look for call records where the CallingIMSI value (the originating number) is the same, but the SwitchNum value (country/region of origin) is different. When you use a JOIN operation with streaming data, the join must provide some limits on how far the matching rows can be separated in time. Because the streaming data is endless, the time bounds for the relationship are specified within the ON clause of the join using the DATEDIFF function. This query is just like a normal SQL join except for the DATEDIFF function. The DATEDIFF function used in this query is specific to Stream Analytics, and it must appear within the ON...BETWEEN clause.



5. Select **Save Query**.

6. Close the Query window to return to the Stream Analytics job page.

8.7.5 Task 5: Start the Stream Analytics job

1. In your **phoneanalysis-asa-job** window, in the **Query** screen in the middle of the window, click on **Start**

2. In the **Start Job** dialog box that opens, click **Now**, and then click **Start**.

   Note: In your **phoneanalysis-asa-job** window, a message appears after a minute that the job has started, and the started field changes to the time started

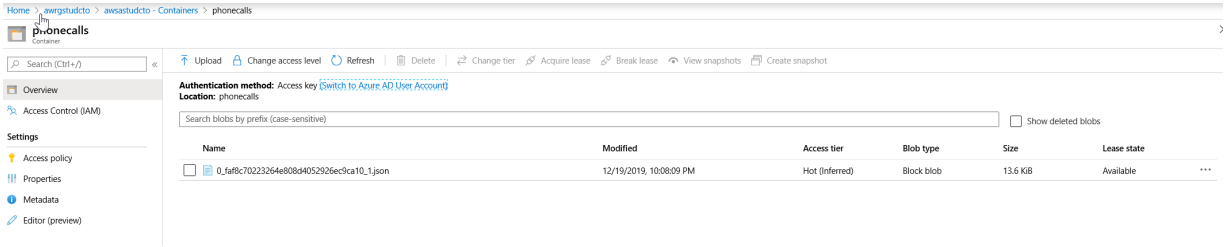
   Note: Leave this running for 2 minutes so that data can be captured.

8.7.6 Task 6: Validate streaming data is collected

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **awsastudxx**, where **xx** are your initials.

2. In the Azure portal, click **Containers** box, and then click on the container named **phonecalls**.

3. Confirm that a JSON file appears, and note the size column.



4. Refresh Microsoft Edge, and when the screen has refreshed note the size of the file

   Note: You could download the file to query the JSON data, you could also output the data to Power BI.

   Result: After you completed this exercise, you have configured Azure Stream Analytics to collect streaming data into an JSON file store in Azure Blob. You have done this with streaming phone call data.

8.8 Close down

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **phoneanalysis-asa-job**.
2. In the **phoneanalysis-asa-job** screen, click on **Stop**. In the **Stop Streaming job** dialog box, click on **Yes**.
3. Close down the Command Prompt application. # DP 200 - Implementing a Data Platform Solution

9 Lab 7 - Orchestrating Data Movement with Azure Data Factory

Estimated Time: 70 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

- **Azure subscription**: If you don't have an Azure subscription, create a free account before you begin.
- **Azure Data Lake Storage Gen2 storage account**: If you don't have an ADLS Gen2 storage account, see the instructions in Create an ADLS Gen2 storage account.
- **Azure Synapse Analytics**: If you don't have a Azure Synapse Analytics account, see the instructions in Create a SQL DW account.

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

9.1 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 Azure Synapse Analytics. The student will also demonstrate how to call a compute resource.

9.2 Lab objectives

After completing this lab, you will be able to:

1. Setup Azure Data Factory
2. Ingest data using the Copy Activity
3. Use the Mapping Data Flow task to perform transformation
4. Perform transformations using a compute resource

9.3 Scenario

You are assessing the tooling that can help with the extraction, load and transforming of data into the data warehouse, and have asked a Data Engineer within your team to show a proof of concept of Azure Data Factory to explore the transformation capabilities of the product. The proof of concept does not have to be related

to AdventureWorks data, and you have given them freedom to pick a dataset of thier choice to showcase the capabilities.

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 lad, you will have:

1. Setup Azure Data Factory
2. Ingested data using the Copy Activity
3. Used the Mapping Data Flow task to perform transformation
4. Performed transformations using a compute resource

> **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.

9.4 Exercise 1: Setup Azure Data Factory

Estimated Time: 15 minutes

Individual exercise

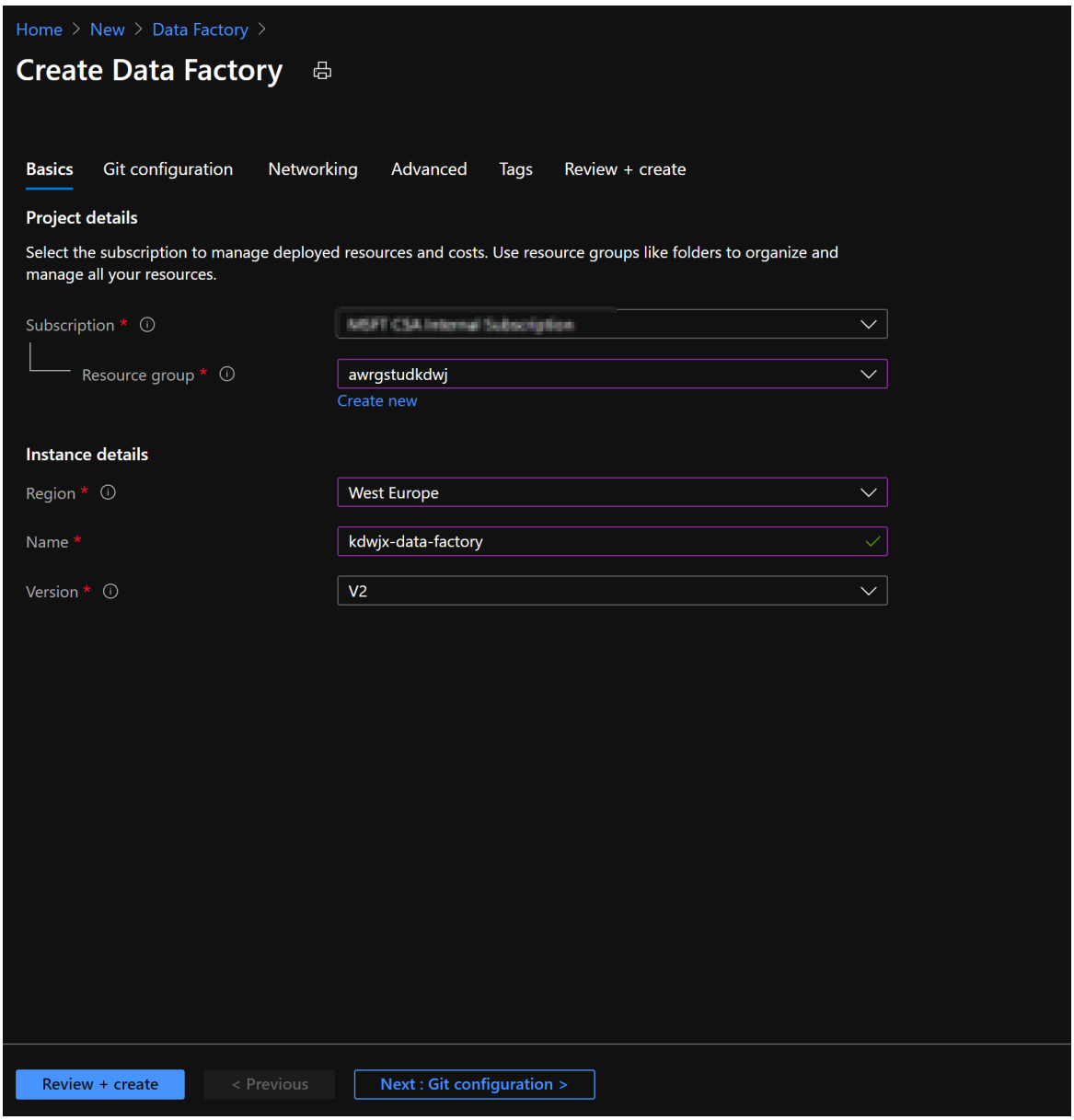
The main task for this exercise are as follows:

1. Setup Azure Data Factory

9.4.1 Task 1: Setting up Azure Data Factory.

Create your data factory: Use the Azure Portal to create your Data Factory.

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



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

3. In the **git configuration** blade **check** Configure git later.
4. Click **review + create** and then select **create**.

   Result: After you completed this exercise, you have created an instance of Azure Data Factory

9.5 Exercise 2: Ingest data using the Copy Activity

Estimated Time: 15 minutes

Individual exercise

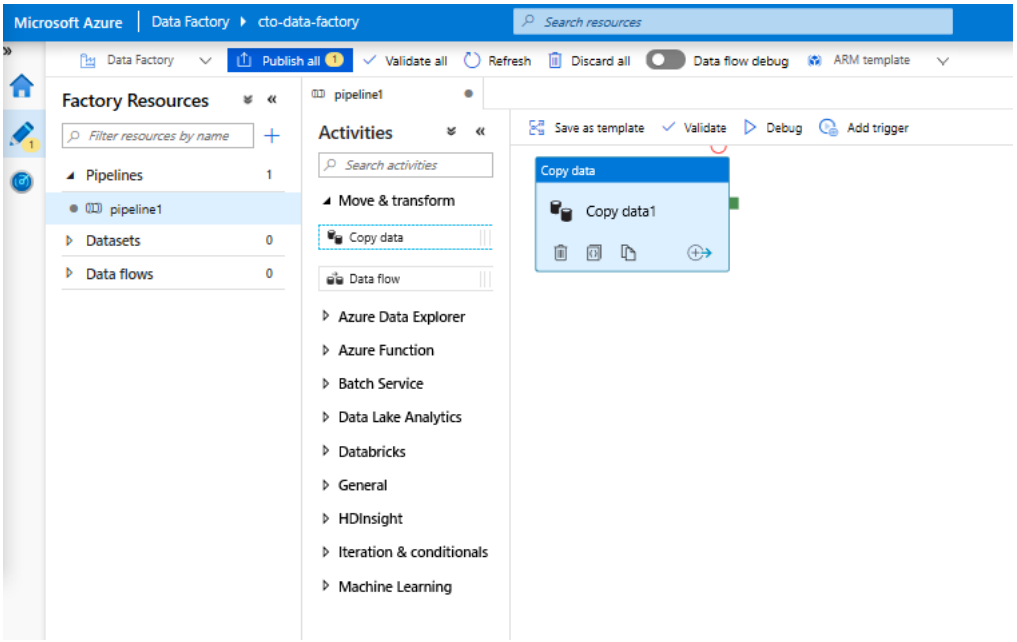
The main tasks for this exercise are as follows:

1. Add the Copy Activity to the designer
2. Create a new HTTP dataset to use as a source
3. Create a new ADLS Gen2 sink
4. Test the Copy Activity

9.5.1 Task 1: Add the Copy Activity to the designer

1. On the deployment successful message, click on the button **Go to resource**.

2. In the xx-data-factory screen, in the middle of the screen, click on the button, **Author & Monitor**
3. **Open the authoring canvas** If coming from the ADF homepage, click on the **pencil icon** on the left sidebar and select the **+ pipeline button** to open the authoring canvas and create a pipeline.
4. **Add a copy activity** In the Activities pane, open the Move and Transform accordion and drag the Copy Data activity onto the pipeline canvas.

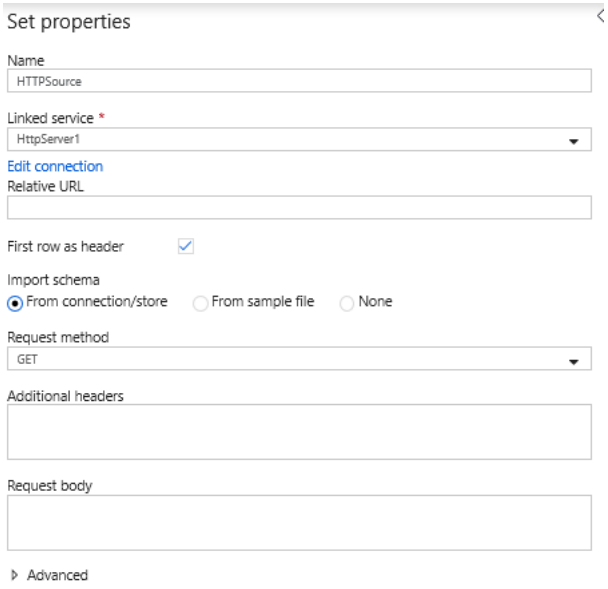


9.5.2 Task 2: Create a new HTTP dataset to use as a source

1. In the Source tab of the Copy activity settings, click **+ New**
2. In the data store list, select the **HTTP** tile and click continue
3. In the file format list, select the **DelimitedText** format tile and click continue
4. In Set Properties blade, give your dataset an understandable name such as **HTTPSource** and click on the **Linked Service** dropdown. If you have not created your HTTP Linked Service, select **New**.
5. In the New Linked Service (HTTP) screen, specify the url of the moviesDB csv file. You can access the data with no authentication required using the following endpoint:

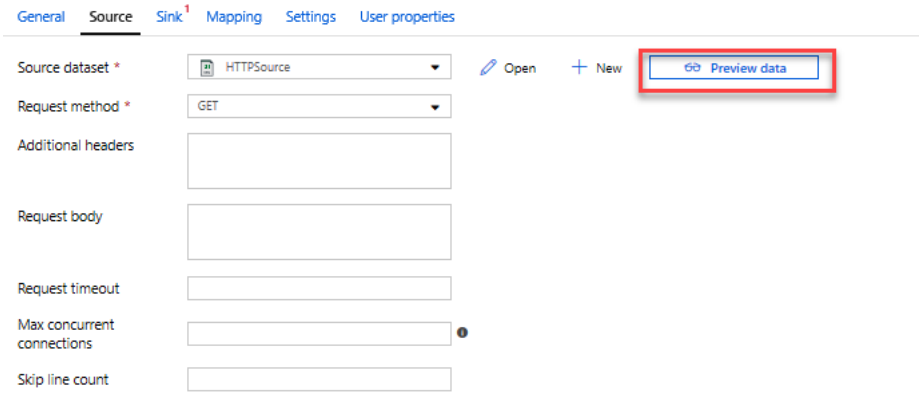
   https://raw.githubusercontent.com/djpmsft/adf-ready-demo/master/moviesDB.csv
6. Place this in the **Base URL** text box.
7. In the **Authentication type** drop down, select **Anonymous**. and click on **Create**.

   - Once you have created and selected the linked service, specify the rest of your dataset settings. These settings specify how and where in your connection we want to pull the data. As the url is pointed at the file already, no relative endpoint is required. As the data has a header in the first row, set **First row as header** to be true and select Import schema from **connection/store** to pull the schema from the file itself. Select **Get** as the request method. You will see the followinf screen



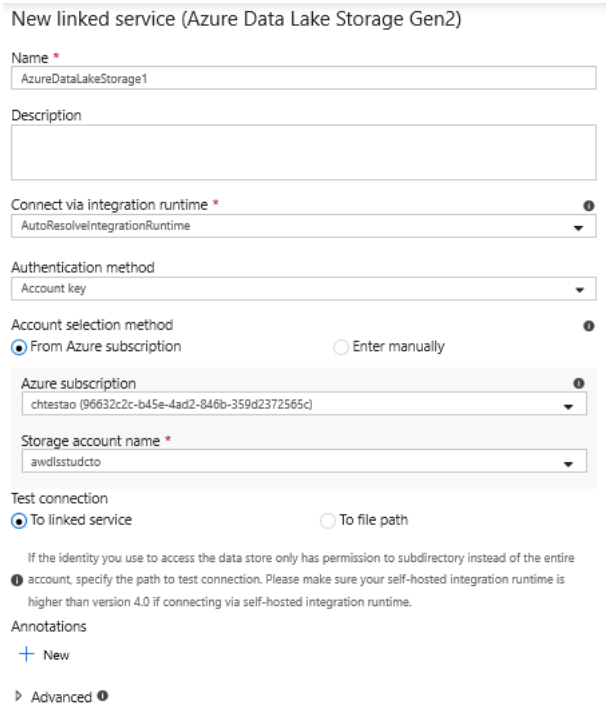
- Click **OK** once completed.

a. To verify your dataset is configured correctly, click **Preview Data** in the Source tab of the copy activity to get a small snapshot of your data.

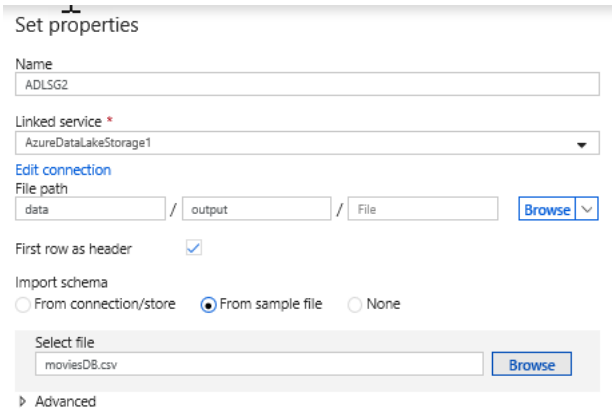


9.5.3 Task 3: Create a new ADLS Gen2 dataset sink

1. Click on the **Sink tab**, and the click + **New**
2. Select the **Azure Data Lake Storage Gen2** tile and click **Continue**.
3. Select the **DelimitedText** format tile and click **Continue**.
4. In Set Properties blade, give your dataset an understandable name such as **ADLSG2** and click on the **Linked Service** dropdown. If you have not created your ADLS Linked Service, select **New**.
5. In the New linked service (Azure Data Lake Storage Gen2) blade, select your authentication method as **Account key**, select your **Azure Subscription** and select your Storage account name of **awdlsstudxx**. You will see a screen as follows:



6. Click on **Create**
7. Once you have configured your linked service, you enter the set properties blade. As you are writing to this dataset, you want to point the folder where you want moviesDB.csv copied to. In the example below, I am writing to folder **output** in the file system **data**. While the folder can be dynamically created, the file system must exist prior to writing to it. Set **First row as header** to be true. You can either Import schema from **sample file** (use the moviesDB.csv file from **Labfiles\Starter\DP-200.7\SampleFiles**)

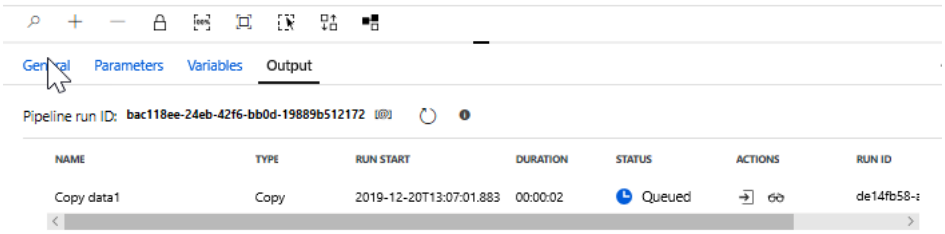


8. Click **OK** once completed.

9.5.4 Task 4: Test the Copy Activity

At this point, you have fully configured your copy activity. To test it out, click on the **Debug** button at the top of the pipeline canvas. This will start a pipeline debug run.

1. To monitor the progress of a pipeline debug run, click on the **Output** tab of the pipeline
2. To view a more detailed description of the activity output, click on the eyeglasses icon. This will open up the copy monitoring screen which provides useful metrics such as Data read/written, throughput and in-depth duration statistics.



3. To verify the copy worked as expected, open up your ADLS gen2 storage account and check to see your file was written as expected

9.6 Exercise 3: Transforming Data with Mapping Data Flow

Estimated Time: 30 minutes

Individual exercise

Now that you have moved the data into Azure Data Lake Store Gen2, you are ready to build a Mapping Data Flow which will transform your data at scale via a spark cluster and then load it into a Data Warehouse.

The main tasks for this exercise are as follows:

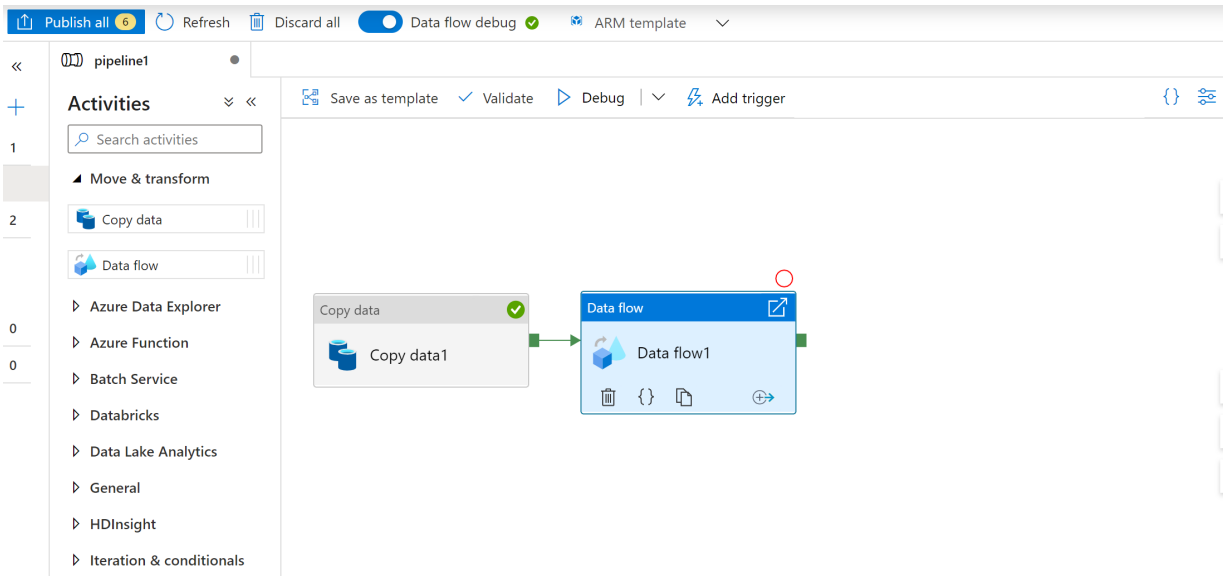
1. Preparing the environment
2. Adding a Data Source
3. Using Mapping Data Flow transformation
4. Writing to a Data Sink
5. Running the Pipeline

9.6.1 Task 1: Preparing the environment

1. **Turn on Data Flow Debug** Turn the **Data Flow Debug** slider located at the top of the authoring module on.

   NOTE: Data Flow clusters take 5-7 minutes to warm up.

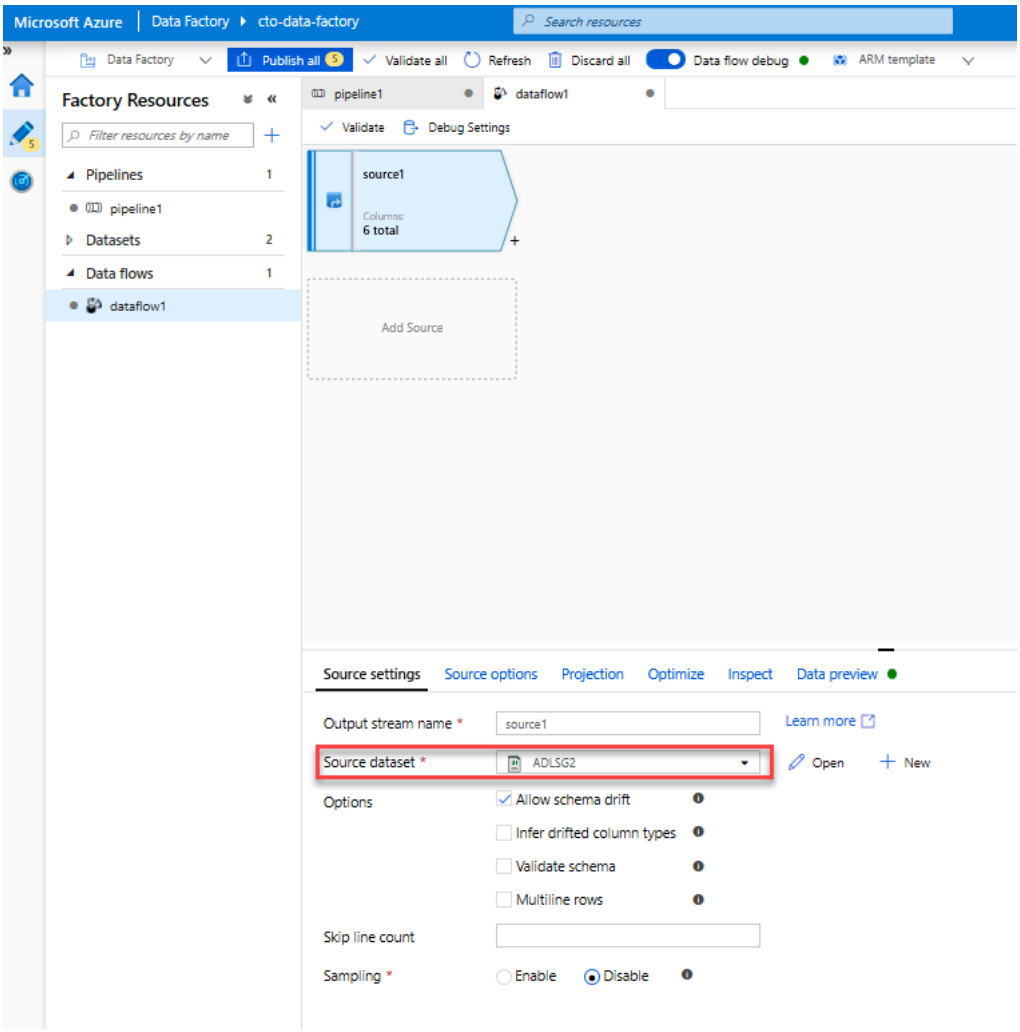
2. **Add a Data Flow activity** In the Activities pane, open the Move and Transform accordion and drag the **Data Flow** activity onto the pipeline canvas.



3. In the setings tab, click + **New** for the variable **Dataflow**

9.6.2 Task 2: Adding a Data Source

1. **Add an ADLS source** Double click on the Mapping Data Flow object in the canvas. Click on the Add Source button in the Data Flow canvas. In the **Source dataset** dropdown, select your **ADLSG2** dataset used in your Copy activity

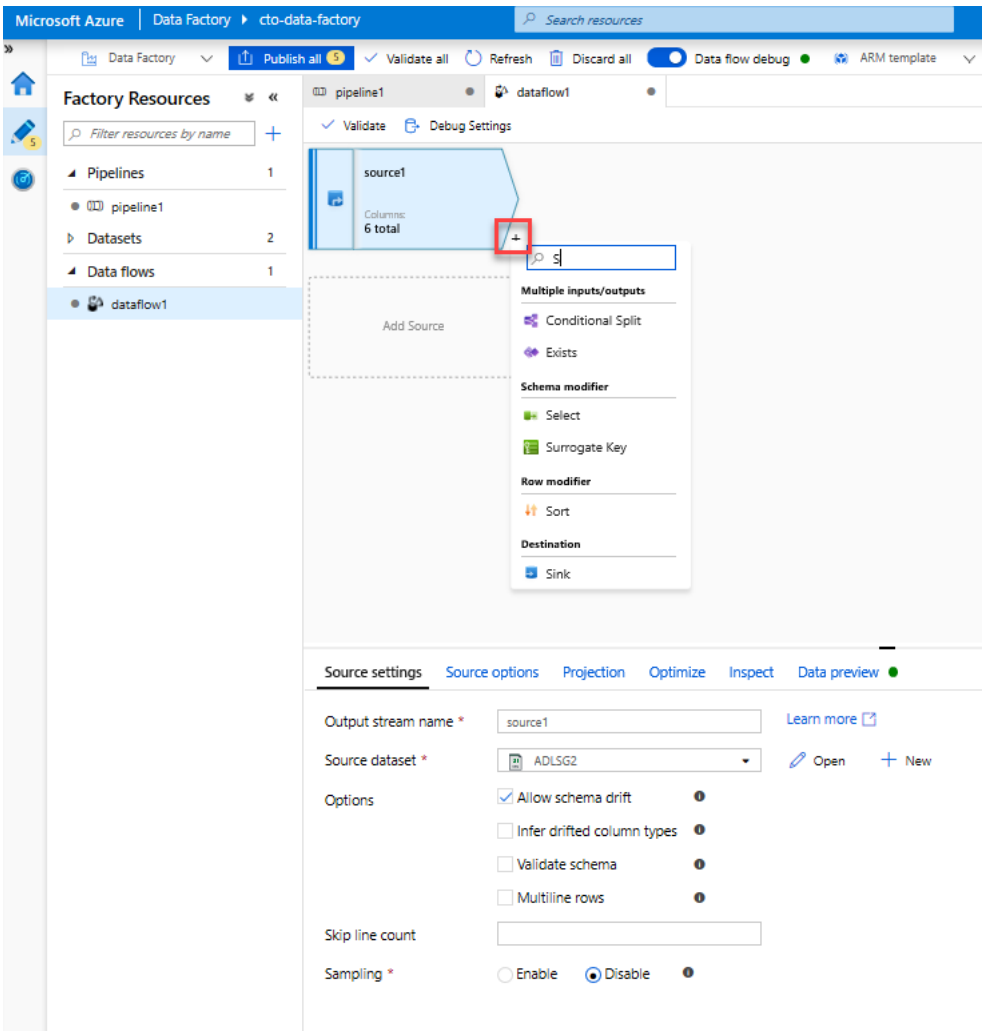


- If your dataset is pointing at a folder with other files, you may need to create another dataset or utilize parameterization to make sure only the moviesDB.csv file is read
- If you have not imported your schema in your ADLS, but have already ingested your data, go to the dataset's 'Schema' tab and click 'Import schema' so that your data flow knows the schema projection.

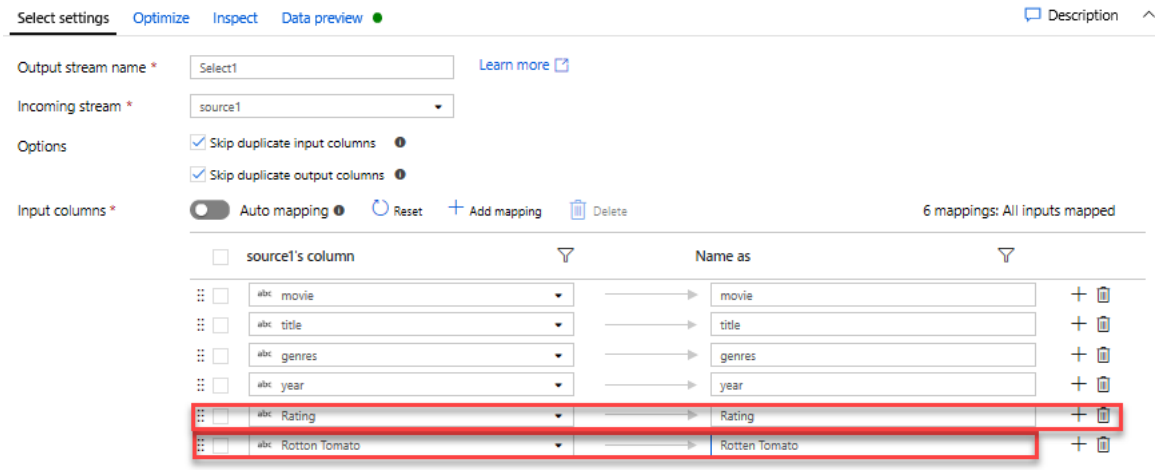
Once your debug cluster is warmed up, verify your data is loaded correctly via the Data Preview tab. Once you click the refresh button, Mapping Data Flow will show calculate a snapshot of what your data looks like when it is at each transformation.

9.6.3 Task 3: Using Mapping Data Flow transformation

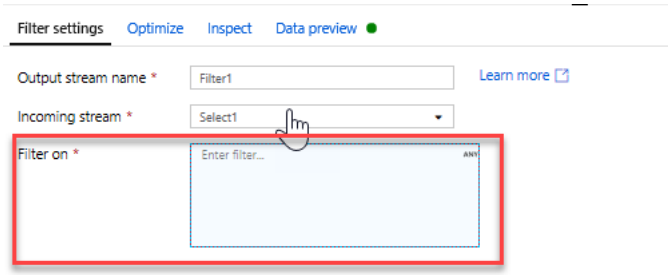
1. **Add a Select transformation to rename and drop a column** In the preview of the data, you may have noticed that the "Rotton Tomatoes" column is misspelled. To correctly name it and drop the unused Rating column, you can add a Select transformation by clicking on the + icon next to your ADLS source node and choosing Select under Schema modifier.



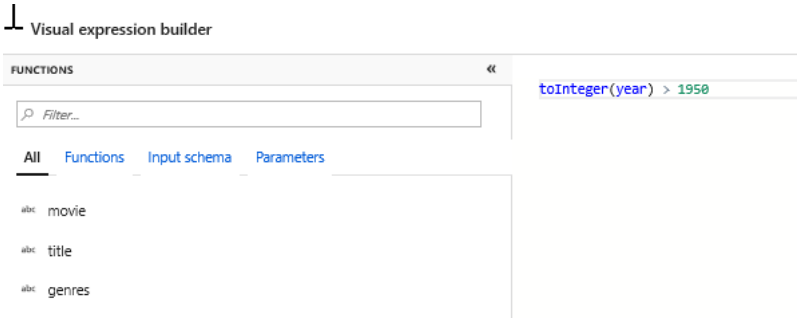
In the **Name** as field, change 'Rotton' to 'Rotten'. To drop the Rating column, hover over it and click on the trash can icon.



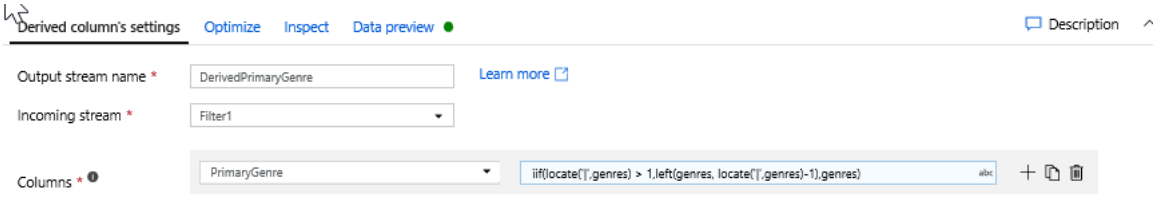
2. **Add a Filter Transformation to filter out unwanted years** Say you are only interested in movies made after 1951. You can add a Filter transformation to specify a filter condition by clicking on the **+ icon** next to your Select transformation and choosing **Filter** under Row Modifier. Click on the **expression box** to open up the Expression builder and enter in your filter condition. Using the syntax of the Mapping Data Flow expression language, **toInteger(year) > 1950** will convert the string year value to an integer and filter rows if that value is above 1950.



You can use the expression builder's embedded Data preview pane to verify your condition is working properly

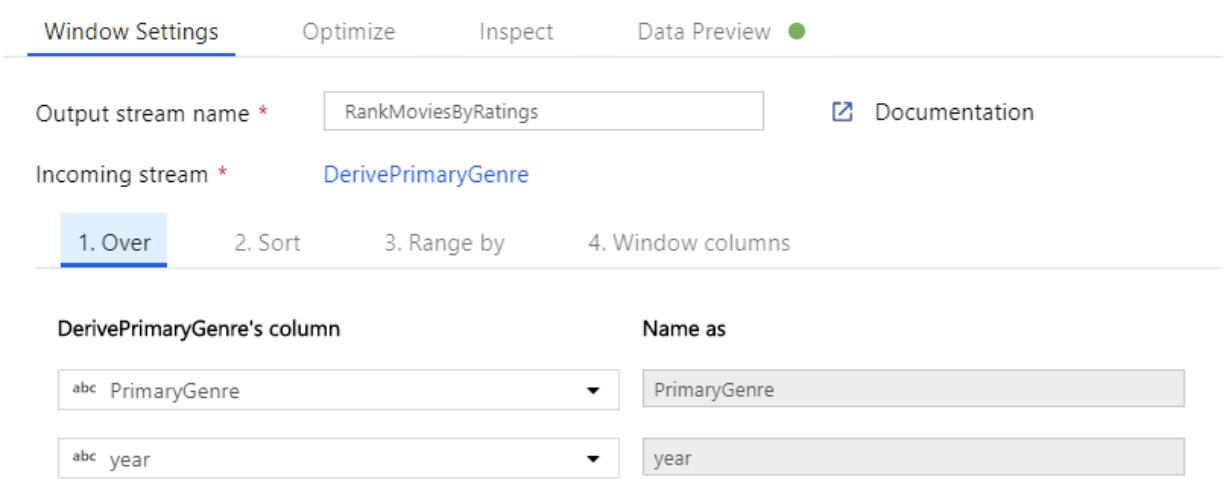


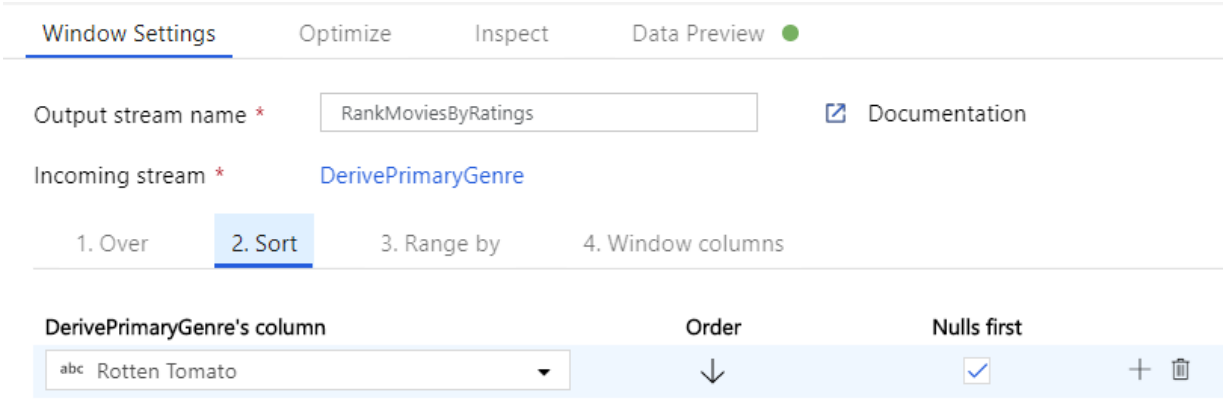
3. **Add a Derive Transformation to calculate primary genre** As you may have noticed, the genres column is a string delimited by a '|' character. If you only care about the *first* genre in each column, you can derive a new column named **PrimaryGenre** via the Derived Column transformation by clicking on the + **icon** next to your Filter transformation and choosing Derived under Schema Modifier. Similar to the filter transformation, the derived column uses the Mapping Data Flow expression builder to specify the values of the new column.

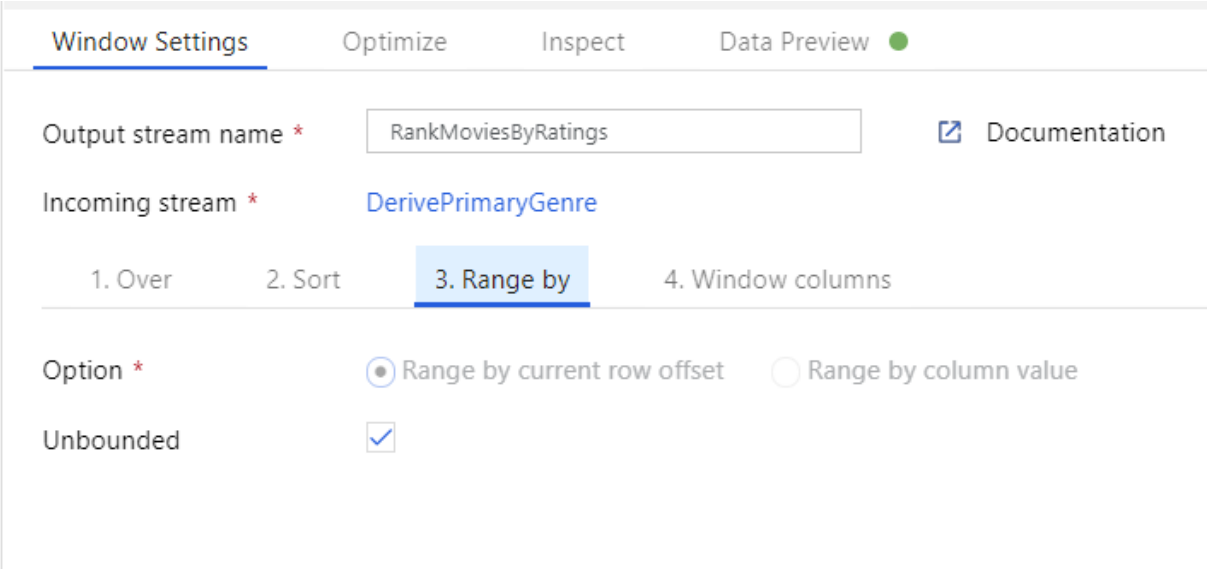


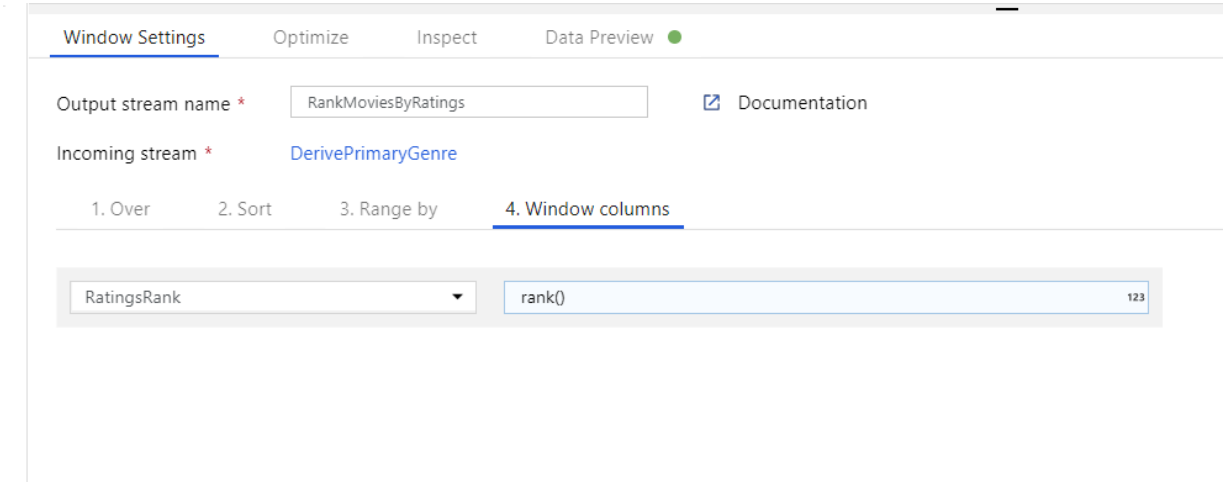
In this scenario, you are trying to extract the first genre from the genres column which is formatted as 'genre1|genre2|...|genreN'. Use the **locate** function to get the first 1-based index of the '|' in the genres string. Using the **iif** function, if this index is greater than 1, the primary genre can be calculated via the **left** function which returns all characters in a string to the left of an index. Otherwise, the PrimaryGenre value is equal to the genres field. You can verify the output via the expression builder's Data preview pane.

4. **Rank movies via a Window Transformation** Say you are interested in how a movie ranks within its year for its specific genre. You can add a Window transformation to define window-based aggregations by clicking on the + **icon** next to your Derived Column transformation and clicking Window under Schema modifier. To accomplish this, specify what you are windowing over, what you are sorting by, what the range is, and how to calculate your new window columns. In this example, we will window over PrimaryGenre and year with an unbounded range, sort by Rotten Tomato descending, a calculate a new column called RatingsRank which is equal to the rank each movie has within its specific genre-year.

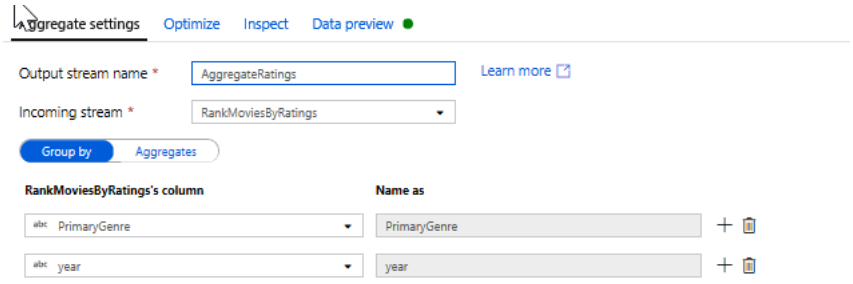

Window Settings
Optimize
Inspect
Data Preview
Output stream name *
RankMoviesByRatings
Documentation
Incoming stream *
DerivePrimaryGenre
1. Over
2. Sort
3. Range by
4. Window columns
DerivePrimaryGenre's column
Name as
PrimaryGenre
PrimaryGenre
year
year


Window Settings
Optimize
Inspect
Data Preview
Output stream name *
RankMoviesByRatings
Documentation
Incoming stream *
DerivePrimaryGenre
1. Over
2. Sort
3. Range by
4. Window columns
DerivePrimaryGenre's column
Order
Nulls first
Rotten Tomato

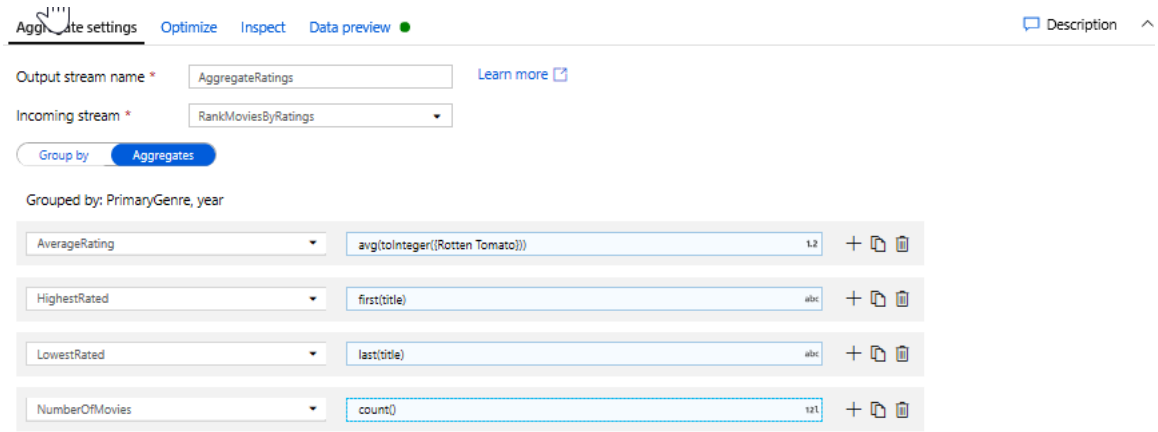

Window Settings
Optimize
Inspect
Data Preview
Output stream name *
RankMoviesByRatings
Documentation
Incoming stream *
DerivePrimaryGenre
1. Over
2. Sort
3. Range by
4. Window columns
Option *
Range by current row offset
Range by column value
Unbounded



5. **Aggregate ratings with an Aggregate Transformation** Now that you have gathered and derived all your required data, we can add an Aggregate transformation to calculate metrics based on a desired group by clicking on the **+ icon** next to your Window transformation and clicking Aggregate under Schema modifier. As you did in the window transformation, lets group movies by PrimaryGenre and year

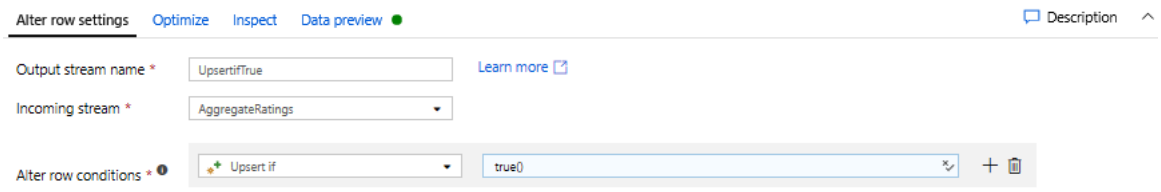


In the Aggregates tab, you can aggregations calculated over the specified group by columns. For every genre and year, lets get the average Rotten Tomatoes rating, the highest and lowest rated movie (utilizing the windowing function) and the number of movies that are in each group. Aggregation significantly reduces the amount of rows in your transformation stream and only propagates the group by and aggregate columns specified in the transformation.



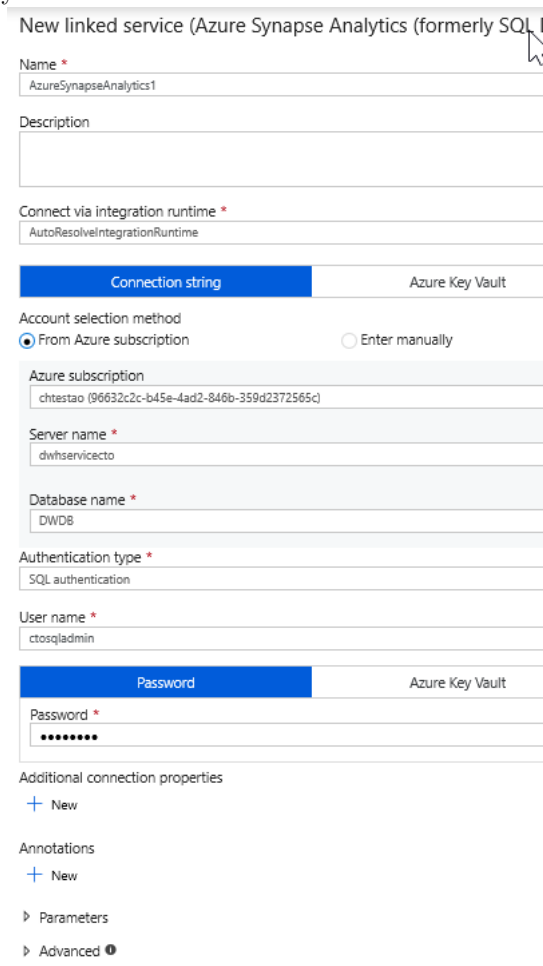
- To see how the aggregate transformation changes your data, use the Data Preview tab

6. **Specify Upsert condition via an Alter Row Transformation** If you are writing to a tabular sink, you can specify insert, delete, update and upsert policies on rows using the Alter Row transformation by clicking on the + icon next to your Aggregate transformation and clicking Alter Row under Row modifier. Since you are always inserting and updating, you can specify that all rows will always be upserted.

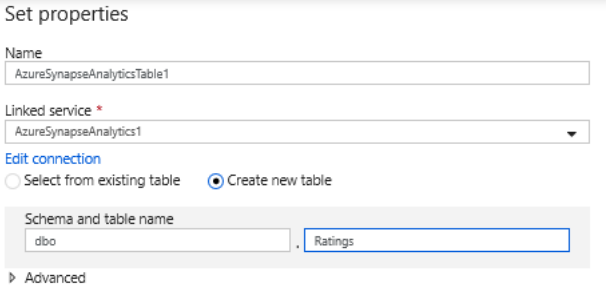


9.6.4 Task 4: Writing to a Data Sink

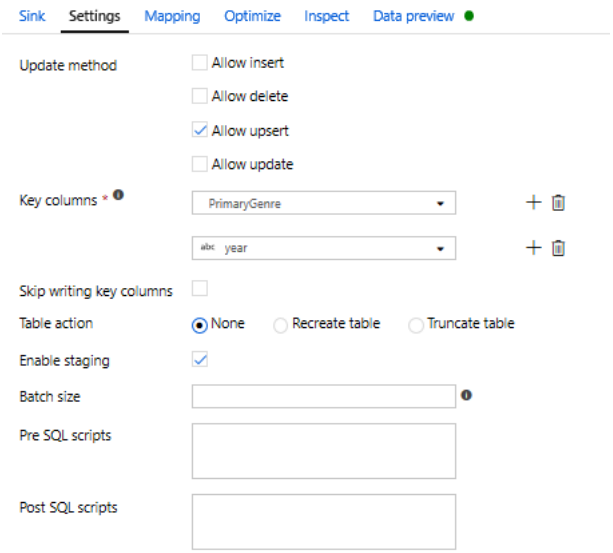
1. **Write to a Azure Synapse Analytics Sink** Now that you have finished all your transformation logic, you are ready to write to a Sink.
   1. Add a **Sink** by clicking on the + **icon** next to your Upsert transformation and clicking Sink under Destination.
   2. In the Sink tab, create a new data warehouse dataset via the + **New button**.
   3. Select **Azure Synapse Analytics** from the tile list.
   4. Select a new linked service and configure your Azure Synapse Analytics connection to connect to the DWDB database created in Module 5. Click **Create** when finished.



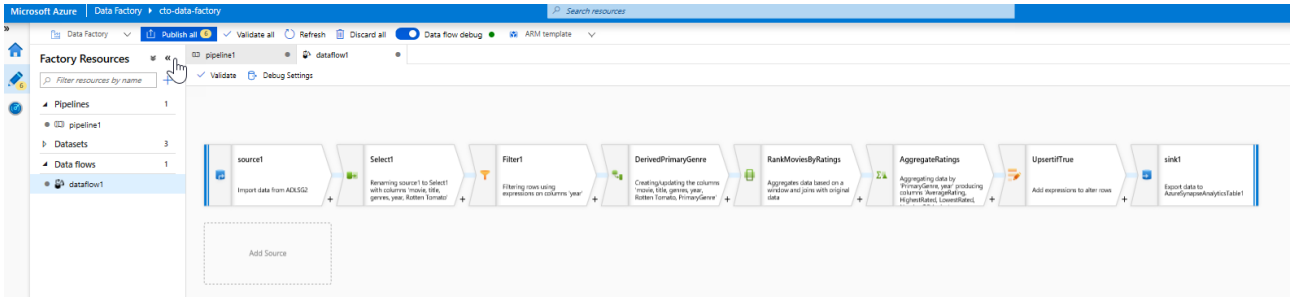
   5. In the dataset configuration, select **Create new table** and enter in the schema of **Dbo** and the table name of **Ratings**. Click **OK** once completed.



   6. Since an upsert condition was specified, you need to go to the Settings tab and select 'Allow upsert'

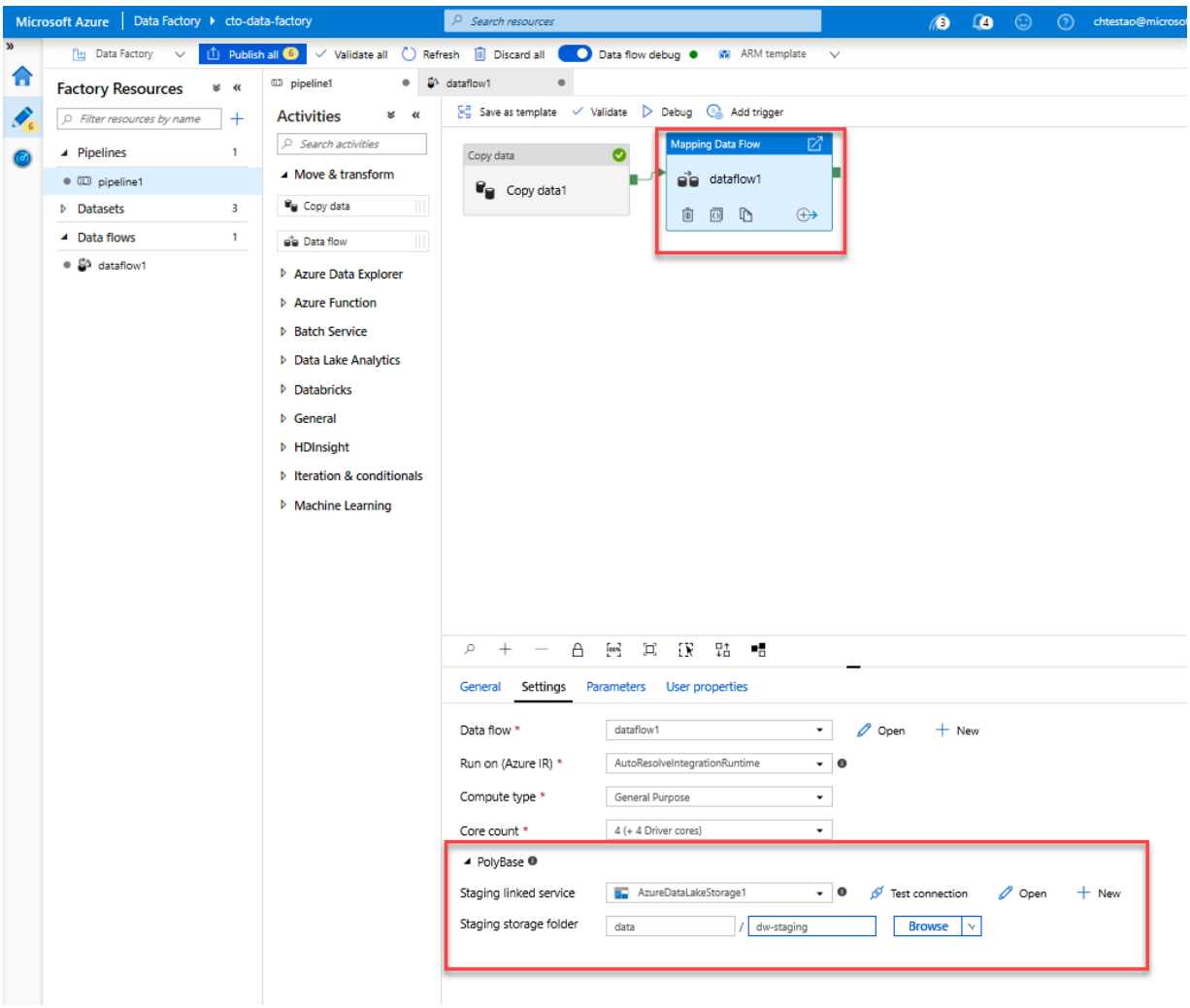


based on key columns PrimaryGenre and year.
At this point, You have finished building your 8 transformation Mapping Data Flow. It's time to run the pipeline and see the results!



9.7 Task 5: Running the Pipeline

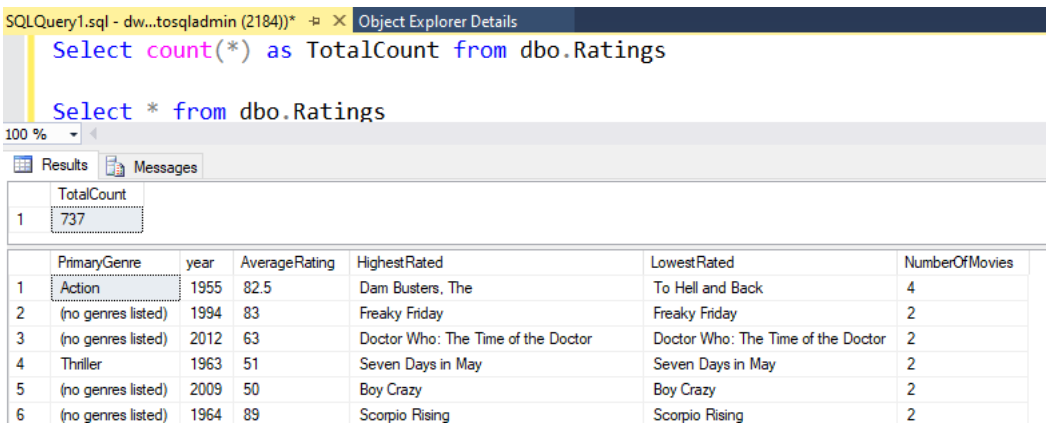
1. Go to the pipeline1 tab in the canvas. Because Azure Synapse Analytics in Data Flow uses PolyBase, you must specify a blob or ADLS staging folder. In the Execute Data Flow activity's settings tab, open up the PolyBase accordion and select your ADLS linked service and specify a staging folder path.



2. Before you publish your pipeline, run another debug run to confirm it's working as expected. Looking at the Output tab, you can monitor the status of both activities as they are running.

3. Once both activities succeeded, you can click on the eyeglasses icon next to the Data Flow activity to get a more in depth look at the Data Flow run.

4. If you used the same logic described in this lab, your Data Flow should will written 737 rows to your SQL DW. You can go into SQL Server Management Studio to verify the pipeline worked correctly and see what got written.



	TotalCount
1	737

	PrimaryGenre	year	AverageRating	HighestRated	LowestRated	NumberOfMovies
1	Action	1955	82.5	Dam Busters, The	To Hell and Back	4
2	(no genres listed)	1994	83	Freaky Friday	Freaky Friday	2
3	(no genres listed)	2012	63	Doctor Who: The Time of the Doctor	Doctor Who: The Time of the Doctor	2
4	Thriller	1963	51	Seven Days in May	Seven Days in May	2
5	(no genres listed)	2009	50	Boy Crazy	Boy Crazy	2
6	(no genres listed)	1964	89	Scorpio Rising	Scorpio Rising	2

9.8 Exercise 4: 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.

9.8.1 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**.

9.8.2 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**

9.8.3 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 exercize.

9.8.4 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.

9.8.5 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.

9.8.6 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.

9.8.7 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.
   # DP 200 - Implementing a Data Platform Solution

10 Lab 8 - Securing Azure Data Platforms

Estimated Time: 75 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 to module 7 has been completed.

Lab files: The files for this lab are located in the *Allfiles\Labfiles\Starter\DP-200.8* folder.

10.1 Lab overview

The students will be able to describe and document the different approaches to security that can be taken to provide defence in depth. This will involve the student documenting the security that has been set up so far in the course. It will also enable the students to identify any gaps in security that may exists for AdventureWorks.

10.2 Lab objectives

After completing this lab, you will be able to:

1. Explain Security
2. Describe key security components
3. Secure Storage Accounts and Data Lake Storage
4. Secure Data Stores
5. Secure Streaming Data

10.3 Scenario

As a senior data engineer within AdventureWorks, you are responsible for ensuring that your data estate is secured. You are performing a security check of your current infrastructure to ensure that you have diligently placed security where it is required. This check should be a holistic check of all the services and data that you have created so far, and an identification of any gaps that there may be in the configuration of the security.

You have also been asked to tighten up the security of the SQL Database DeptDatabasesxx and have been asked to setup auditing against the database so that you can monitor access to the database. Furthermore, you have learned that that the Manage permission for your event hub is not restrictive enough, and you want to remove this permission.

At the end of this lad, you will have:

1. Explained Security
2. Described key security components
3. Secured Storage Accounts and Data Lake Storage
4. Secured Data Stores
5. Secured Streaming Data

> **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.

10.4 Exercise 1: An introduction to security

Estimated Time: 15 minutes

Group exercise

The main task for this exercise are as follows:

1. Security as a layered approach.
2. The instructor will discuss the findings with the group.

10.4.1 Task 1: Security as a layered approach.

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab08-Ex01.docx** from the **Allfiles\Labfiles\Starter\DP-200.8** folder.

2. From the course content, case study and the scenarios taken in the course so far, spend **10 minutes** in a group identifying the layers of security that you have impacted so far to secure AdventureWorks in the labs. Find three examples.

10.4.2 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 contains at least three examples of how you have implemented security at Adventureworks and which layer of security you have impacted.

10.5 Exercise 2: Key security components

Estimated Time: 10 minutes

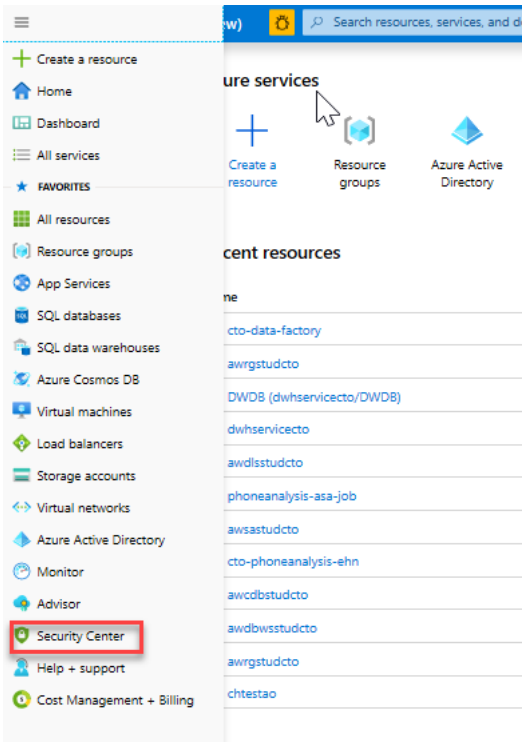
Individual exercise

The main tasks for this exercise are as follows:

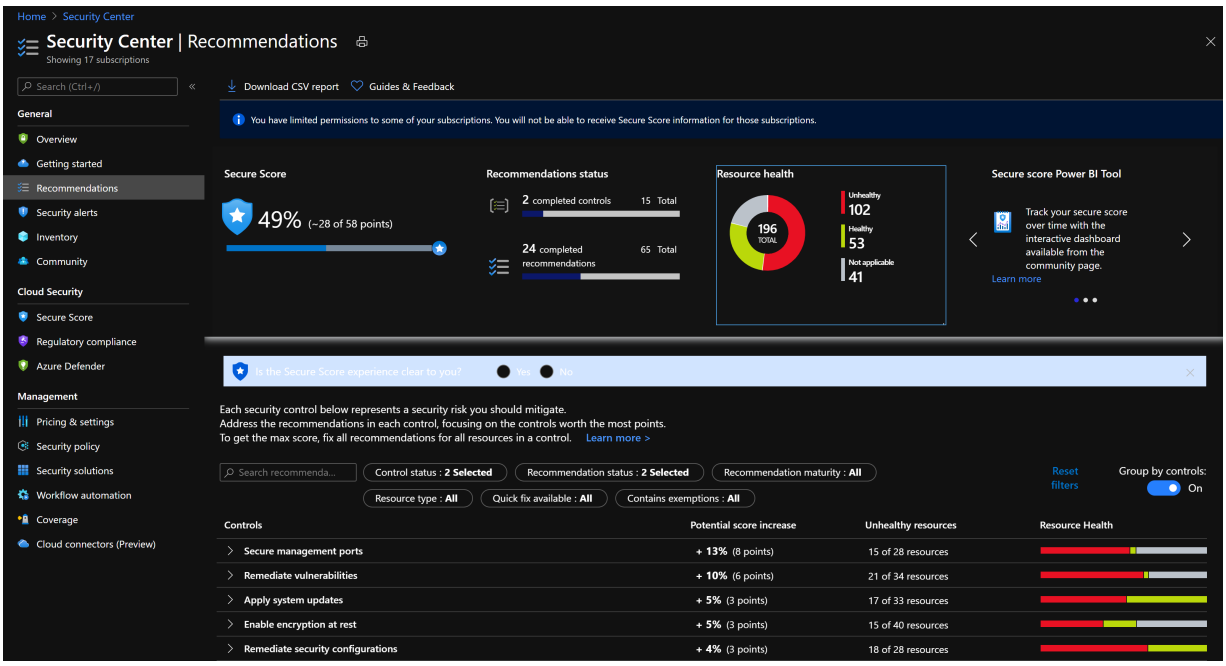
1. Assessing Data and Storage Security Hygiene

10.5.1 Task 1: Assessing Data and Storage Security Hygiene.

1. In the Azure portal tab, click **Security Center**.



2. In the Security Center - Overview screen, click **Recommendations**.



3. Identify the top two key data and storage components that require attention.

    1. **Answers may vary**_____
    2. **Answers may vary**_____

    Result: After you completed this exercise, you have learned where you can look to identify any data and storage security weaknesses that is in your Azure subscription.

10.6 Exercise 3: Securing Storage Accounts and Data Lake Storage

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Determining the appropriate security approach for Azure Blob
2. Discuss the findings with the Instructor

10.6.1 Task 1: Determining the appropriate security approach for Azure Blob

1. You have been approached by your in-house web developer to help give access to a third party web design company to the web images that are in the awsastudxx storage account. As a senior data engineer within AdventureWorks, what steps would you need to take to ensure this can happen while applying the correct due diligence.
2. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab08-Ex03.docx** from the **Allfiles\Labfiles\Starter\DP-200.8** folder.

10.6.2 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 contains the steps that you would take to provide secure access to a Blob storage account to a third-party web development company.

10.7 Exercise 4: Securing Data Stores

Estimated Time: 15 minutes

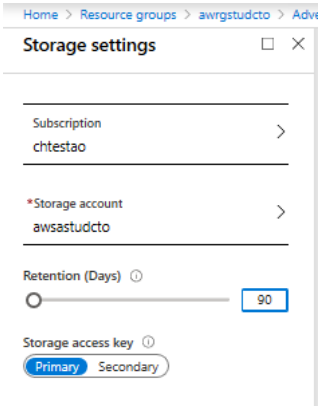
Individual exercise

The main tasks for this exercise are as follows:

1. Enabling Auditing
2. Query the Database
3. View the Audit log

10.7.1 Task 1: Enabling Auditing

1. In the Azure portal, click **Resource groups**, and then click awrgstudxx, and then click on the sqlservicexx and navigate to **AdventureworksLT (sqlservicekdwj/AdventureworksLT)**.
2. In the AdventureworksLT (sqlservicekdwj/AdventureworksLT) screen, click on the **Auditing** blade.
3. Under **Auditing**, click on the **ON** button.
4. Check the **Storage** box and then click on **Storage Details - Configure**.
5. In the **Storage Setting** screen, click **Subscription - change storage subscription**, and then click your subscription.
6. In the **Storage Setting** screen, click **Storage Settings - Configure required settings**. In the **Choose storage account** screen, click **awsastudxx**
7. In the **Retention Days** text box, type **90**, and then click on **OK**.



8. Click on **Save**.

10.7.2 Task 2: Query the database

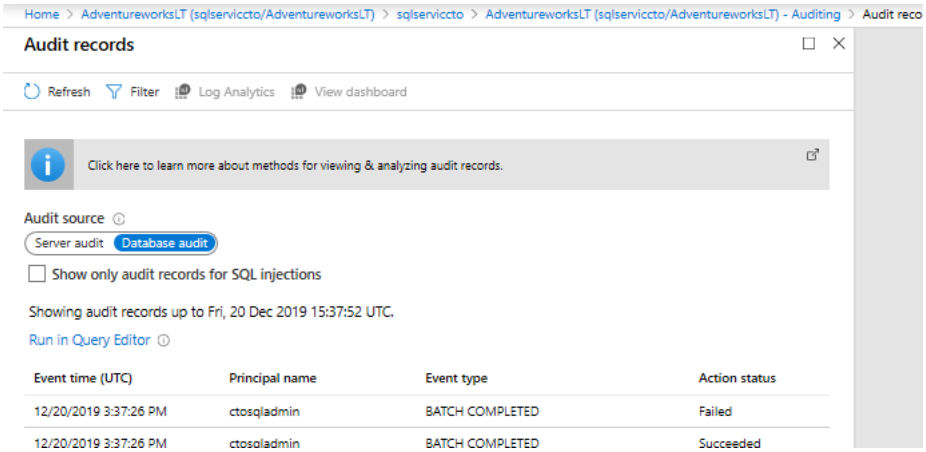
1. In the Azure portal, click **Resource groups**, and then click awrgstudxx, and then click on the sqlservicexx and navigate to **AdventureworksLT (sqlservicekdwj/AdventureworksLT)**.
2. Navigate to the **query editor**
3. In the **SQL server authentication** pane that shows up log in with the following details:
   - Username: **xxsqladmin**
   - Password: **P@ssw0r**
4. Click **ok**

   Note: An error message is returned as the password is incorrect. Type in the correct password of **P@Ssw0rd**.
5. Type in the correct password of **Pa55w.rd**
6. In **query editor**, expand **AdventureWorksLT**, and then expand **Tables**.
7. Right click [SalesLT].[Customers] and then click **Select Top 1000 Rows**

10.7.3 Task 2: View the Audit Log

1. Return to the Azure Portal. In the AdventureWorksLT (sqlservicexx/AdventureWorksLT) - Auditing screen, click on **View Audit Logs**

2. Note in the **Audit records** log file the **Failed Authentication** record. Close down the **Audit records** screen



Result: After you completed this exercise, you have enabled database auditing and verified that the auditing works.

10.8 Exercise 5: Securing Streaming Data

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Changing Event Hub Permissions

10.8.1 Task 1: Changing Event Hub Permissions

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **xx-phoneanalysis-ehn**, where **xx** are your initials
2. In the Azure portal, in the **xx-phoneanalysis-ehn**, where **xx** are your initials. Scroll to the bottom of the window, and click on **xx-phoneanalysis-eh** event hub.
3. To grant access to the event hub, click **Shared access policies**.
4. Under the **xx-phoneanalysis-eh - Shared access policies** screen, click on **phoneanalysis-eh-sap**.
5. Click on the checkbox next to the **Manage** permissions to remove it, and then click **Save**.
6. In the Azure portal, in the blade, click **Home**,

   Result: After you completed this exercise, you modified the security of an Event Hub Shared Access Policy. # DP 200 - Implementing a Data Platform Solution

11 Lab 9 - Monitoring and Troubleshooting Data Storage and Processing

Estimated Time: 75 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 to module 7 has been completed.

Lab files: The files for this lab are located in the *Allfiles\Labfiles\Starter\DP-200.9* folder.

11.1 Lab overview

The students will be able to define a broad monitoring solution that can help them monitor issues that can occur in their data estate. The student will then experience common data storage issues and data processing issue that can occur in cloud data solution. Finally they will implement a disaster recovery approach for a Data Platform technology.

11.2 Lab objectives

After completing this lab, you will be able to:

1. Explain the monitoring capabilities that are available
2. Troubleshoot common data storage issues
3. Troubleshoot common data processing issues
4. Manage disaster recovery

11.3 Scenario

As the Senior Data Engineer at AdventureWorks you have been tasked with defining the standard operating procedures for monitoring that data estate within the organization. You will start by defining the monitoring tools that will be used to support the approach.

You will then explore some of the common data storage and data processing issues that can occur during the normal operation of your infrastructure.

There are concerns around the recovery of the Products database that is stored in the awcdbstudxx Cosmos DB. The IS department has asked you to provide high level steps that would be taken in the event that the products database has become unavailable through an accidental deletion or removal of the database.

At the end of this lad, you will have:

1. Explained the monitoring capabilities that are available
2. Troubleshoot common data storage issues
3. Troubleshoot common data processing issues
4. Managed disaster recovery

11.4 Exercise 0: Issue review

> **IMPORTANT**: In this lab, you will refer to the issue(s) that you have encountered in any provisioning or configuration tasks in all of the labs in the document located at *\Labfiles\DP-200-Issues-Doc.docx*. Open this document up ready for the following exercises.

11.5 Exercise 1: Explain the monitoring capabilities that are available

Estimated Time: 15 minutes

Group exercise

The main task for this exercise are as follows:

1. Defining a corporate monitoring approach.
2. The instructor will discuss the findings with the group.

11.5.1 Task 1: Defining a corporate monitoring approach.

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab09-Ex01.docx** from the **Allfiles\Labfiles\Starter\DP-200.** folder.
2. Spend **10 minutes** in a group discussing and identifying the monitoring tools that would be the most useful tool for you within your organization. Find two examples and outline your justification.

11.5.2 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 identifies the monitoring tools that would be the most useful tool for you within your organization.

11.6 Exercise 2: Troubleshoot common data storage issues

Estimated Time: 20 minutes

Group exercise

The main tasks for this exercise are as follows:

1. Assessing issues that are data storage issues.
2. The instructor will discuss the findings with the group.

11.6.1 Task 1: Assessing Data and Storage Security Hygiene.

1. In the **\Labfiles\DP-200-Issues-Doc.docx**_document, share your findings and work with the group to identify which of the issues are data storage issues.
2. Work with the group to see if there are **common data storage issue** that the group experienced while setting up the various data platforms.
3. As a group select two data storage issues that you have identified, and log them into the document **DP-200-Lab09-Ex02.docx**.

11.6.2 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 lists two data storage issues.

11.7 Exercise 3: Troubleshoot common data processing issues

Estimated Time: 20 minutes

Group exercise

The main tasks for this exercise are as follows:

1. Assessing issues that are data processing issues.
2. The instructor will discuss the findings with the group.

11.7.1 Task 1: Assessing Data and Storage Security Hygiene.

1. Review with the Group the questions outlined in **DP-200-Lab09-Ex03.docx**
2. Work with the group to see if there are **common data processing issues** that the group experienced while setting up the various data platforms.
3. As a group select two data storage issues that you have identified, and log them into the document **DP-200-Lab09-Ex03.docx**.

11.7.2 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 lists two data processing issues.

11.8 Exercise 4: Manage disaster recovery

Estimated Time: 20 minutes

Group exercise

The main tasks for this exercise are as follows:

1. Manage Disaster Recovery.
2. The instructor will discuss the findings with the group.

11.8.1 Task 1: Manage Disaster Recovery

1. From the lab virtual machine, start **Microsoft Word**, and open up the file **DP-200-Lab09-Ex04.docx** from the **Allfiles\Labfiles\Starter\DP-200.9** 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.

11.8.2 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 outlines the high level steps required to restore the products database. # DP 200 - Implementing a Data Platform Solution

12 Lab 6 - Performing Real-Time Analytics with Stream Analytics

Estimated Time: 60 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-200.6* folder.

12.1 Lab overview

The students will be able to describe what data streams are and how event processing works and choose an appropriate data stream ingestion technology for the AdventureWorks case study. They will provision the chosen ingestion technology and integrate this with Stream Analytics to create a solution that works with streaming data.

12.2 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

12.3 Scenario

As part of the digital transformation project, you have been tasked by the CIO to help the marketing departments become more productive with aspects of their work. Over the last few years the marketing department has been using Twitter to amplify marketing message around the bicycle products that are sold.

Whilst the department can provide reach numbers post campaign, they are unable to understand who is interacting with their campaigns in real-time, as the volumes are difficult to track manually. As a result, they would like to implement a system that can track in real-time who is responding to their campaign.

At the end of this lad, you will have:

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

   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.

12.4 Exercise 1: Explain data streams and event processing

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.

12.4.1 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.

12.4.2 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 .

12.5 Exercise 2: Data Ingestion with Event Hubs.

Estimated Time: 15 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Create and configure an Event Hub Namespace.
2. Create and configure an Event Hub.
3. Configure Event Hub security.

12.5.1 Task 1: Create and configure an Event Hub Namespace.

1. In the Azure portal, select **+ Create a resource**, type **Event Hubs**, and then select **Event Hubs** from the resulting search. Then select **Create**.
2. In the Create Namespace blade, type out the following options, then click **Create**:
   - **Name**: xx-socialtwitter-eh, where xx are your initials
   - **Pricing Tier**: Standard
   - **Subscription**: Your subscription
   - **Resource group**: awrgstudxx
   - **Location**: select the location closest to you
   - Leave other options to their default settings

   Note: The creation of the Event Hub Namespace takes approximately 1 minute.

12.5.2 Task 2: Create and configure an Event Hub

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **awdlsstudxx**, where **xx** are your initials
2. Click on **xx-socialtwitter-eh**, where **xx** are your initials.
3. In the **xx-socialtwitter-eh** screen, click on **+ Event Hubs**.
4. Provide the name **socialtwitter-eh**, and then select **Create**.

   Note: You will receive a message stating that the Event Hub is created after about 10 seconds

12.5.3 Task 3: Configure Event Hub security

1. In the Azure portal, in the **xx-socialtwitter-eh**, where **xx** are your initials. Scroll to the bottom of the window, and click on **socialtwitter-eh** event hub.
2. To grant access to the event hub, click **Shared access policies**.
3. Under the **socialtwitter-eh - Shared access policies** screen, create a policy with **Manage** permissions by selecting **+ Add**. Give the policy the name of **socialtwitter-eh-sap** , check **Manage**, and then click **Create**.

4. Click on your new policy **socialtwitter-eh-sap** after it has been created, and then select the copy button for the **CONNECTION STRING - PRIMARY KEY** and paste the CONNECTION STRING - PRIMARY KEY into Notepad, this is needed later in the exercise.
5. Close down the Event hub screens in the portal

   Result: After you completed this exercise, you have created an Azure Event Hub within an Event Hub Namespace and set the security for the Event Hub that can be used to provide access to the service.

12.6 Exercise 3: Processing Data with Stream Analytics Jobs

Estimated Time: 30 minutes

Individual exercise

The main tasks for this exercise are as follows:

1. Create a Twitter developer account
2. Configure the Twitter access keys
3. Configure and start the Twitter client application
4. Provision a Stream Analytics job.
5. Specify the a Stream Analytics job input.
6. Defining a Stream Analytics query.
7. Specify the a Stream Analytics job output.
8. Modify a Stream Analytics query.
9. Start the Stream Analytics job.
10. Validate streaming data is collected

12.6.1 Task 1: Create a Twitter developer account.

1. Open a new tab in your browser and go to https://developer.twitter.com/en/apps/ and login with your twitter account to set up a Twitter application.
2. In **Apps** page, to the right, click on the button **Create an app**.
3. In the "Please apply for a twitter account", click on **Apply**.
4. In the "User profile" page, confirm your twitter account is the correct account to associate with a developer account, and click on **Continue**.
5. In the "Account details" page, under "Who are you requesting access for?" click **I am requesting access for my own personal use**. Under "Account name", type **learn how to code as *twitter_name***. Finally, select **Primary country of operation** as the country where you reside. Click on **Continue**.

   Note: please remember to observe all applicable laws and ethics for your country of operation
6. In the "Tell us about your project" page, choose an option for "What use case(s) are you interested in?". For the purpose of this course select **Student**. In the "Describe in your own words what you are building", spend 300 words to explain why this is being used. below is an example:

   Note: the questions and options in this screen can change.

```
>1. I'm using Twitter's APIs to learn how to embed twitter statements within a learning application. th
>2.  I plan to analyze Tweets to understand how to count and read streaming tweets in real time as part
>3. I do not intend on interacting with other peoples tweets in the form of retweeting or liking conten
>4. Both individual and aggregate tweets will be analysed in the application
```

1. Finally, select the option for the question "Will your product, service, or analysis make Twitter content or derived information available to a government entity?" and then click on **Continue**.
2. Read and agree to the Terms of Service by clicking **Accept** if you are happy to do so. Then select the checkboxes as you see fit. Click on **Submit application**

3. Check your inbox for the verification email, click on the **confirm your email** within the email to complete the process. You will be directed back to twitter to a Welcome screen.

12.6.2 Task 2: Configure the Twitter access keys

1. If you are not at the Twitter welcome screen, Open a new tab in your browser and go to https://developer.twitter.com and login with your twitter account.
2. In the "Welcome" or "Apps" page, click on the button **Create an app**.
3. In the "Create an App" page, provide the following information and click on **Create an App**:
   - **name**: xx-social-app, where xx are your initials
   - **Application description**: add a description such as "used to collect and aggregate tweets"
   - **website URL** a personal address that you can use for the application.
   - **Tell us how this app will be used**: write text that outlines the app is a demo app to learn how to use streaming data
4. Review the Developer terms and click on **Create**. A new page will appear with the title of you application name.
5. Click on the page tab named **Keys and tokens**
6. Under Access token & access token secret, Click on the **Create** button.

   Note: An authorized access token and secret will be generated for your account and the current application.
7. Copy the following keys to Notepad file you had previously opened:
   - Consumer Key (API Key)
   - Consumer Secret (API Secret)
   - Access Token
   - Access Token Secret

12.6.3 Task 3: Configure and start the Twitter client application

1. Double click the **TwitterWPFClient.exe** application in the **Allfiles\Labfiles\Starter\DP-200.6\Twitter client\TwitterWPFClient\TwitterWPFClient** location. You can also download the TwitterWPFClient here.
2. Enter your data from Notepad into the TwitterWPFClient.exe application.
   - **Twitter Consumer Key (API Key)**: Your twitter consumer key
   - **Twitter Consumer Secret (API Secret)**: Your twitter consumer secret
   - **Twitter Access Token**: Your twitter access token
   - **Twitter Access Secret**: Your twitter access secret
   - **Azure EventHub Name**: socialtwitter-eh.
   - **Azure Event Hub Connection string**: Endpoint=sb://xx-socialtwitter-eh.servicebus.windows.net/;SharedAcc social-eh-sap;SharedAccessKey=<Paste in the CONNECTION STRING - PRIMARY KEY from Notepad>

     Important: IT IS IMPORTANT TO REMOVE ;EntityPath=socialtwitter-eh AT THE END OF THE STRING.
   - **Search Groups** define which keywords you want to determine sentiment for. An example could be "#Brexit", or brexit, or you can add multiple keywords separated by a comma.
3. Click the **Play** button in the **TwitterWPFClient.exe** application, wait until you see tweets appearing in the console and leave the application running.

   Note: The **TwitterWPFClient.exe** application capture the date, search term sentiment score, twitter name and tweet details for the search term defined.
4. Keep all applications running

12.6.4 Task 4: Provision a Stream Analytics job.

1. Go back to the Azure portal, select + **Create a resource**, type **STREAM**, and then click the **Stream Analytics Job**, and then click **CREATE**.
2.
3. In the **New Stream Analytics job** screen, fill out the following details and then click on **Create**:

   - **Job name**: socialtwitter-asa-job.
   - **Subscription**: select your subscription
   - **Resource group**: awrgstudxx
   - **Location**: choose a location nearest to you.
   - Leave other options to their default settings

   Note: You will receive a message stating that the Stream Analytics job is created after about 10 seconds

12.6.5 Task 5: Specify the a Stream Analytics job input.

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **awdlsstudxx**, where **xx** are your initials
2. Click on **socialtwitter-asa-job**.
3. In your **socialtwitter-asa-job** Stream Analytics job window, click **Inputs**.
4. In the **Inputs** screen, click + **Add stream input**, and then click **Event Hubs**.
5. In the Event Hub screen, type in the following values and click the **Save** button.

   - **Input alias**: Enter a name for this job input as TwitterStream.
   - **Select Event Hub from your subscriptions**: checked
   - **Subscription**: Your subscription name
   - **Event Hub Namespace**: xx-socialtwitter-eh
   - **Event Hub Name**: Use existing named socialtwitter-eh
   - **Event Hub Policy Name**: socialtwitter-eh-sap
   - Leave remaining options to their default

6. Once completed, the TwitterStream Input job will appear under the input window. Close the input widow to return to the Streaming Analytics Job Page

12.6.6 Task 6: Specify the a Stream Analytics job output.

1. In your **socialtwitter-asa-job** Stream Analytics job window, click **Outputs**.
2. In the **Outputs** screen, click + **Add**, and then click **Blob Storage**.
3. In the **Blob storage** window, type or select the following values in the pane:

- **Output alias**: Output
- **Select Event Hub from your subscriptions**: checked
- **Subscription**: Your subscription name
- **Storage account**: awsastudxx, where xx is your initials
- **Container**: Use existing and select tweets

1. Leave the rest of the entries as default values. Finally, click **Save***.
2. Close the output screen to return to the Stream Analytics job page

12.6.7 Task 7: Defining a Stream Analytics query.

1. In your **socialtwitter-asa-job** window, in the **Query** screen in the middle of the window, click on ** Edit query**
2. Replace the following query in the code editor:

```
SELECT
    *
INTO
```

```
    [YourOutputAlias]
FROM
    [YourInputAlias]
```

3. Replace with

```
SELECT
 [Topic]
,[SentimentScore]
,[created_at]
,[Author]
,[text]
FROM [TwitterStream]
```

4. Select Save.
5. Close the Query window to return to the Stream Analytics job page.

12.6.8 Task 8: Defining a Stream Analytics query.

1. In your **socialtwitter-asa-job** window, in the **Query** screen in the middle of the window, click on **Edit query**
2. Replace the following query in the code editor:

```
SELECT
 [Topic]
,[SentimentScore]
,[created_at]
,[Author]
,[text]
FROM [TwitterStream]
```

3. Replace with

```
SELECT
 [Topic]
,[SentimentScore]
,[created_at]
,[Author]
,[text]
INTO [Outputs]
FROM [TwitterStream]
```

4. Select Save.
5. Close the Query window to return to the Stream Analytics job page.

12.6.9 Task 9: Start the Stream Analytics job

1. In your **socialtwitter-asa-job** window, in the **Query** screen in the middle of the window, click on **Start**
2. In the **Start Job** dialog box that opens, click **Now**, and then click **Start**.

   Note: In your **socialtwitter-asa-job** window, a message appears after a minute that the job has started, and the started field changes to the time started

   Note: Leave this running for 2 minutes so that data can be captured.

12.6.10 Task 10: Validate streaming data is collected

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **awsastudxx**, where **xx** are your initials.
2. In the Azure portal, click **Blobs** screen.
3. In the Azure portal, in the **awsastudxx - Blobs** screen, click on the **tweets** item in the list.
4. Confirm that a JSON file appears, and note the size column.

5. Refresh Microsoft Edge, and when the screen has refreshed note the size of the file

   Note: You could download the file to query the JSON data, you could also output the data to Power BI.

   Result: After you completed this exercise, you have configured Azure Stream Analytics to collect streaming data into an JSON file store in Azure Blob. You have done this with streaming twitter data.

12.7 Close down

1. In the Azure portal, in the blade, click **Resource groups**, and then click **awrgstudxx**, and then click on **socialtwitter-asa-job**.
2. In the **socialtwitter-asa-job** screen, click on **Stop**. In the **Stop Streaming job** dialog box, click on **Yes**.
3. Close down the **TwitterWPFClient.exe** application. # DP 200 - Implementing a Data Platform Solution

13 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.

13.1 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.

13.2 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

13.3 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 lad, you will have:

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

13.4 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.

13.4.1 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.

13.4.2 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 .

13.5 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

13.5.1 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.

13.5.2 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**.

13.5.3 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**.

13.5.4 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**.

13.5.5 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**.

13.5.6 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**

13.5.7 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:

13.5.8 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

13.5.9 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.

13.6 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.

13.6.1 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**.

13.6.2 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**

13.6.3 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 exercize.

13.6.4 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.

13.6.5 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.

13.6.6 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.

13.6.7 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.