dp-200-03_instructions.md

DP 200 - Implementing a Data Platform Solution

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.

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.

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

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.

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.

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.

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.

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

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.

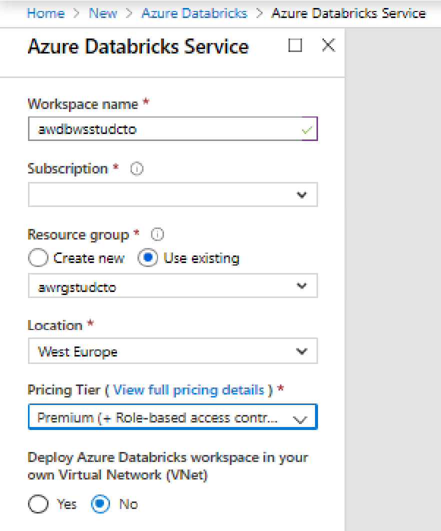
    - **Subscription**: the name of the subscription you are using in this lab

    - **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)**.

    - **Deploy Azure Databricks workspace in your Virtual Network**: **No**.



Home > New > Azure Databricks > Azure Databricks Service

Azure Databricks Service

Workspace name *
awdbwsstudcto

Subscription *

Resource group *
○ Create new ● Use existing
awrgstudcto

Location *
West Europe

Pricing Tier (View full pricing details) *
Premium (+ Role-based access contr...

Deploy Azure Databricks workspace in your own Virtual Network (VNet)
○ Yes ● No

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).

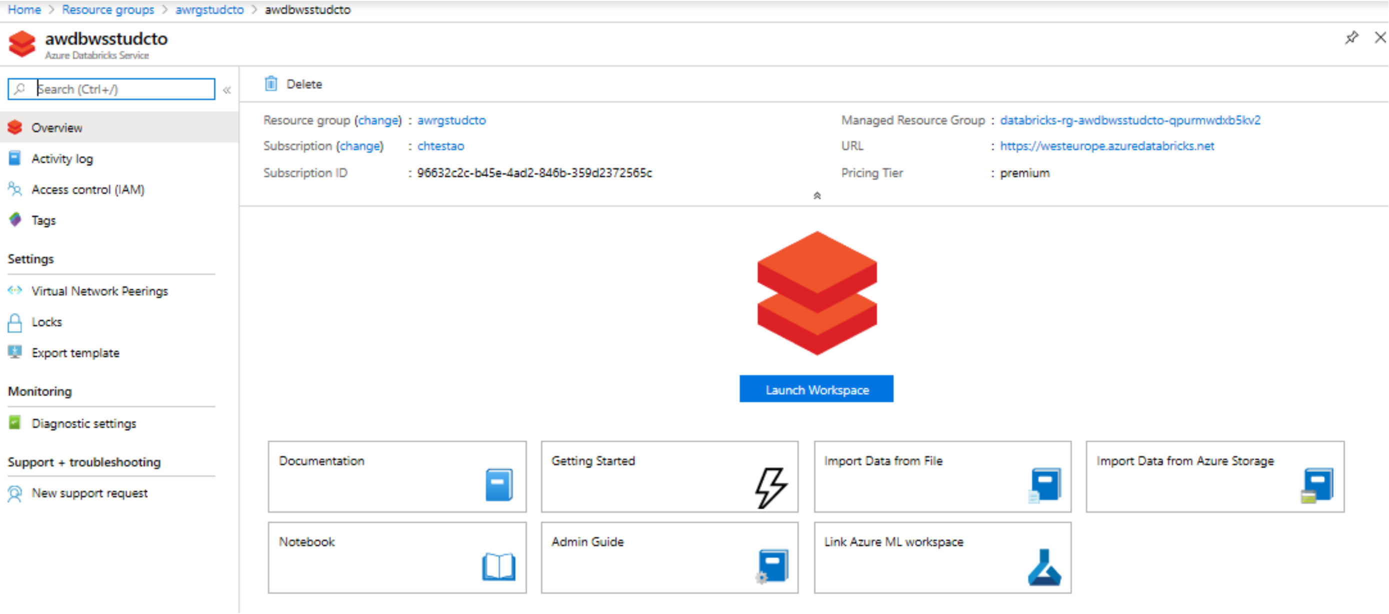
Task 2: Open Azure Databricks.

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.



Home > Resource groups > awrgstudcto > awdbwsstudcto

awdbwsstudcto
Azure Databricks Service

Delete

Resource group (change) : awrgstudcto
Subscription (change) : chtestao
Subscription ID : 96632c2c-b45e-4ad2-846b-359d2372565c

Managed Resource Group : databricks-rg-awdbwsstudcto-qpurmwdxb5kv2
URL : https://westeurope.azuredatabricks.net
Pricing Tier : premium

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

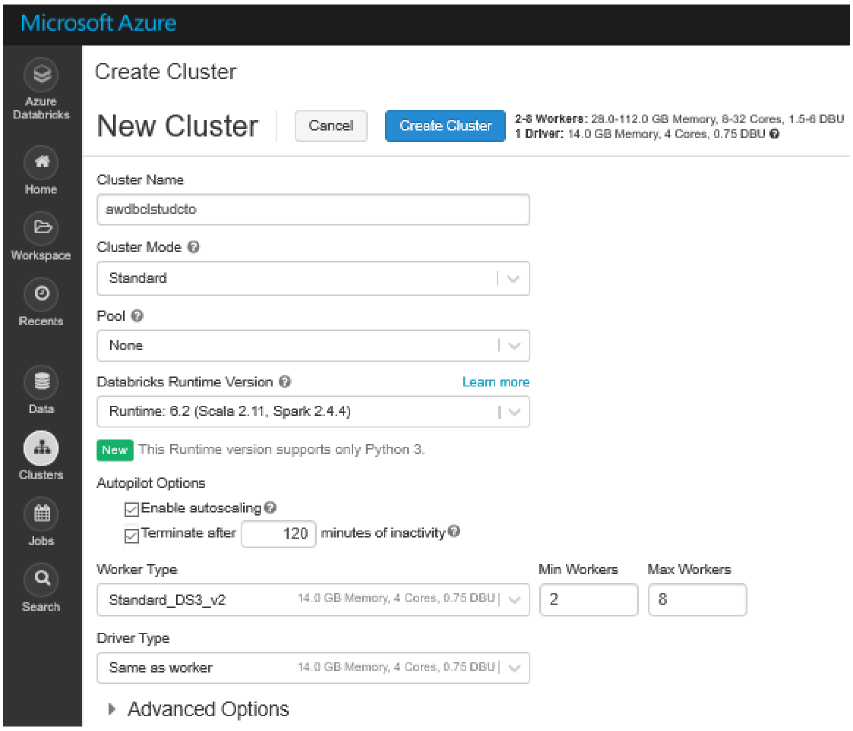
1. In the Azure portal, in the **awdbwsstudxx** screen, click on the button **Launch Workspace**.

> **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: 6.3 (Scala 2.11, Spark 2.4.4)**
    - 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.
    - Leave all the remaining options to their current settings.



Create Cluster

New Cluster

2-8 Workers: 28.0-112.0 GB Memory, 8-32 Cores, 1.5-6 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU

Cluster Name: awdbclstudcto

Cluster Mode: Standard

Pool: None

Databricks Runtime Version: Runtime: 6.2 (Scala 2.11, Spark 2.4.4)

This Runtime version supports only Python 3.

Autopilot Options

☑ Enable autoscaling

☑ Terminate after 120 minutes of inactivity

Worker Type: Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU

Min Workers: 2

Max Workers: 8

Driver Type: Same as worker 14.0 GB Memory, 4 Cores, 0.75 DBU

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**.

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.

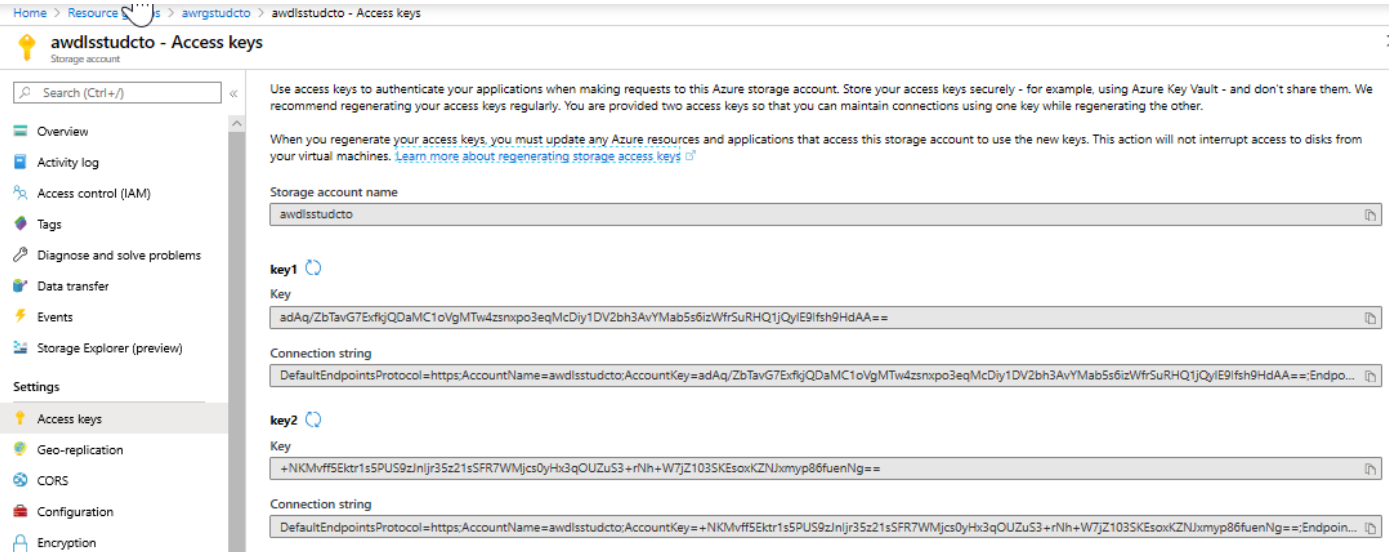
Task 1: Confirm the creation of the Databricks cluster

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.

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.



awdlsstudcto - Access keys

Storage account

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure Key Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. Learn more about regenerating storage access keys

Storage account name: awdlsstudcto

key1

Key: adAq/ZbTavG7ExfkjQDaMC1oVgMTw4zsnxpo3eqMcDiy1DV2bh3AvYMab5s6izWfrSuRHQ1jQylE9lfsh9HdAA==

Connection string: DefaultEndpointsProtocol=https;AccountName=awdlsstudcto;AccountKey=adAq/ZbTavG7ExfkjQDaMC1oVgMTw4zsnxpo3eqMcDiy1DV2bh3AvYMab5s6izWfrSuRHQ1jQylE9lfsh9HdAA==;Endpo...

key2

Key: +NKMvff5Ektr1s5PUS9zJnljr35z21sSFR7WMjcs0yHx3qOUZuS3+rNh+W7jZ103SKEsoxKZNJxmyp86fuenNg==

Connection string: DefaultEndpointsProtocol=https;AccountName=awdlsstudcto;AccountKey=+NKMvff5Ektr1s5PUS9zJnljr35z21sSFR7WMjcs0yHx3qOUZuS3+rNh+W7jZ103SKEsoxKZNJxmyp86fuenNg==;Endpoin...

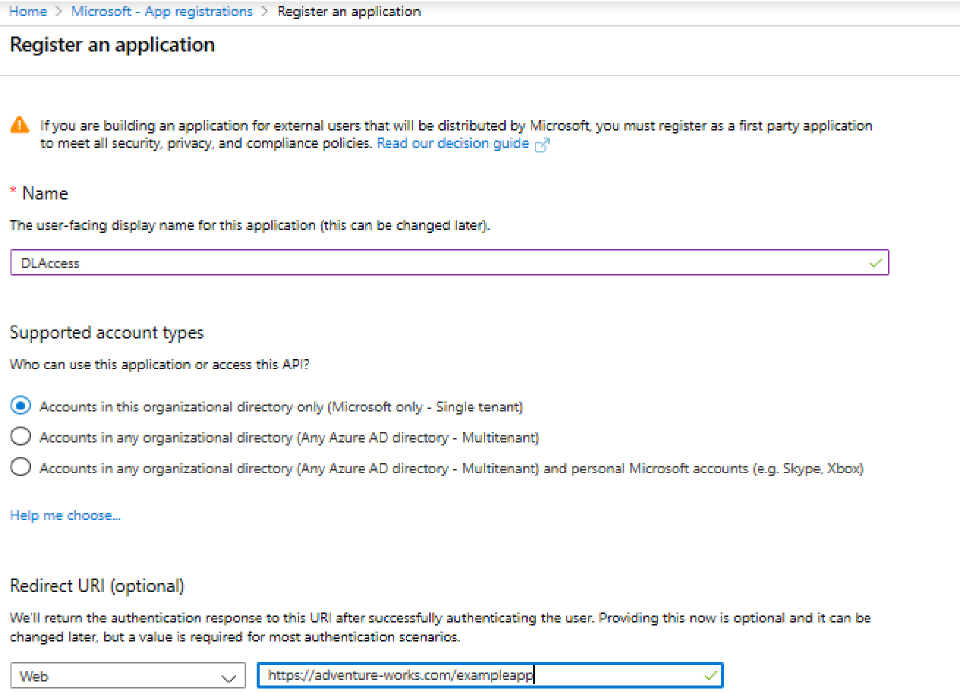
Task 3: Enable your Databricks instance to access the Data Lake Gen2 Store.

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 **https://adventure-works.com/exampleapp** for the application value. After setting the values.


Home > Microsoft - App registrations > Register an application

Register an application

If you are building an application for external users that will be distributed by Microsoft, you must register as a first party application to meet all security, privacy, and compliance policies. Read our decision guide

* Name

The user-facing display name for this application (this can be changed later).

DLAccess ✓

Supported account types

Who can use this application or access this API?

● Accounts in this organizational directory only (Microsoft only - Single tenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant)

○ Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

Help me choose...

Redirect URI (optional)

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

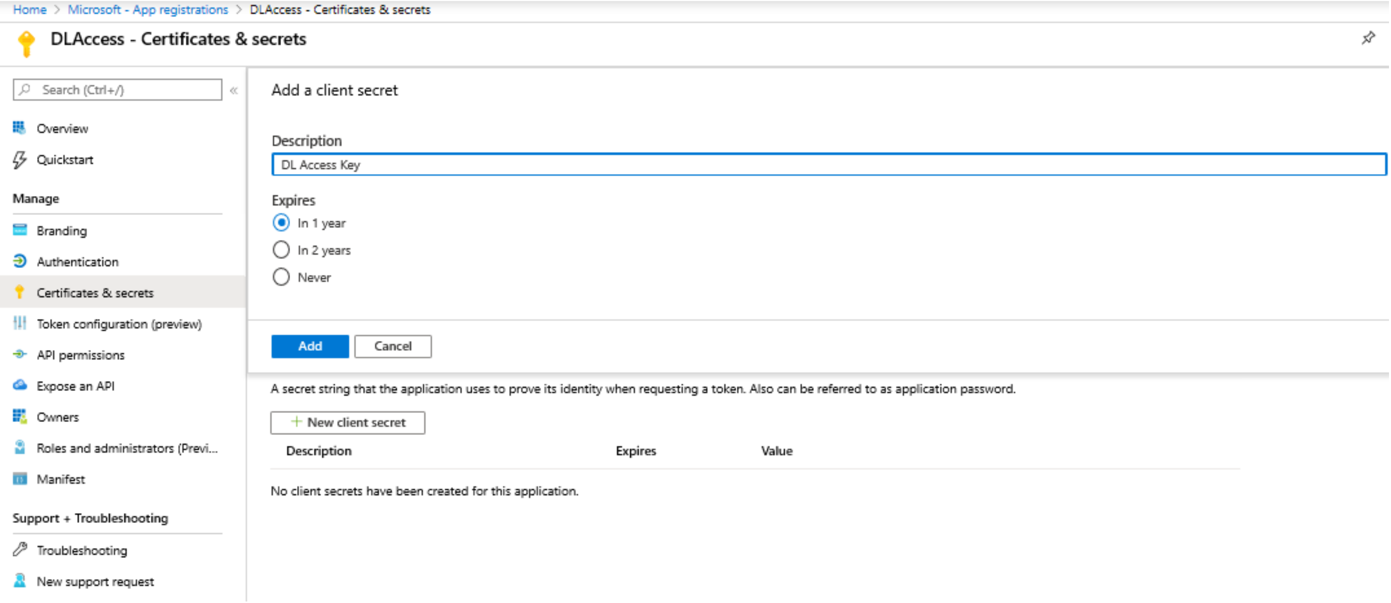
Web | https://adventure-works.com/exampleapp ✓

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**.


Home > Microsoft - App registrations > DLAccess - Certificates & secrets
DLAccess - Certificates & secrets
Search (Ctrl+/)
Overview
Quickstart
Manage
Branding
Authentication
Certificates & secrets
Token configuration (preview)
API permissions
Expose an API
Owners
Roles and administrators (Previ...
Manifest
Support + Troubleshooting
Troubleshooting
New support request
Add
Cancel
+ New client secret

Add a client secret

Description: DL Access Key

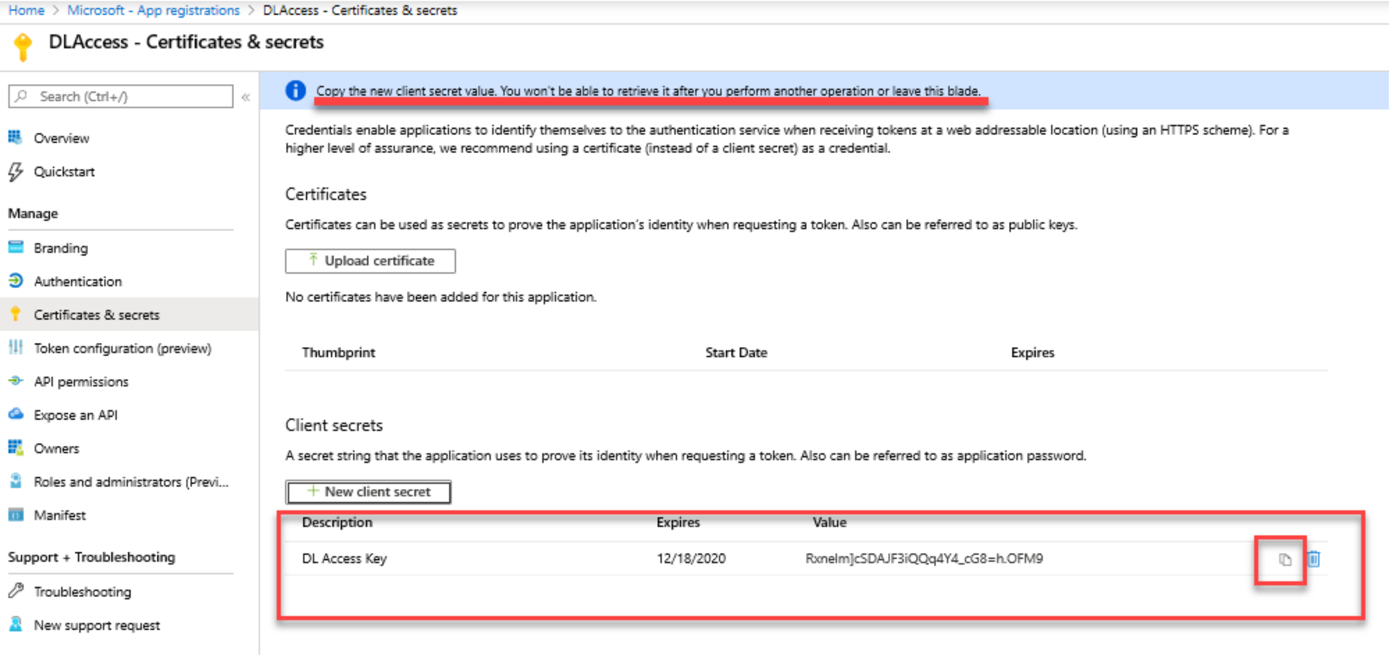
Expires
● In 1 year
○ In 2 years
○ Never

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value

No client secrets have been created for this application.

> **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



Copy the new client secret value. You won't be able to retrieve it after you perform another operation or leave this blade.

Credentials enable applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.

Certificates

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

No certificates have been added for this application.

Thumbprint	Start Date	Expires

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value
DL Access Key	12/18/2020	Rxnelm]cSDAJF3iQQq4Y4_cG8=h.OFM9

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**.

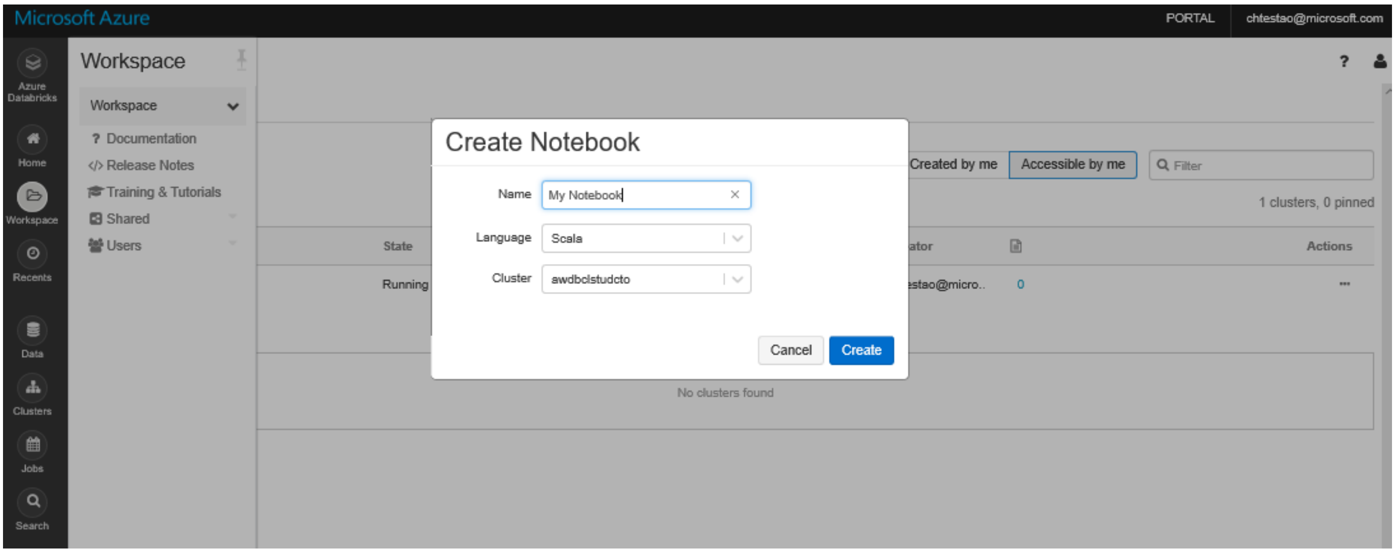
16. In the Azure portal, click the **Home** hyperlink, and then click the **Azure Active Directory** icon, Note **your role**. If you have the User role, you must make sure that non-administrators can register applications.
17. Click **Users**, and then click **User settings** in the **Users - All users** 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**

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**

   Create Notebook

   Name: My Notebook

   Language: Scala

   Cluster: awdbclstudcto

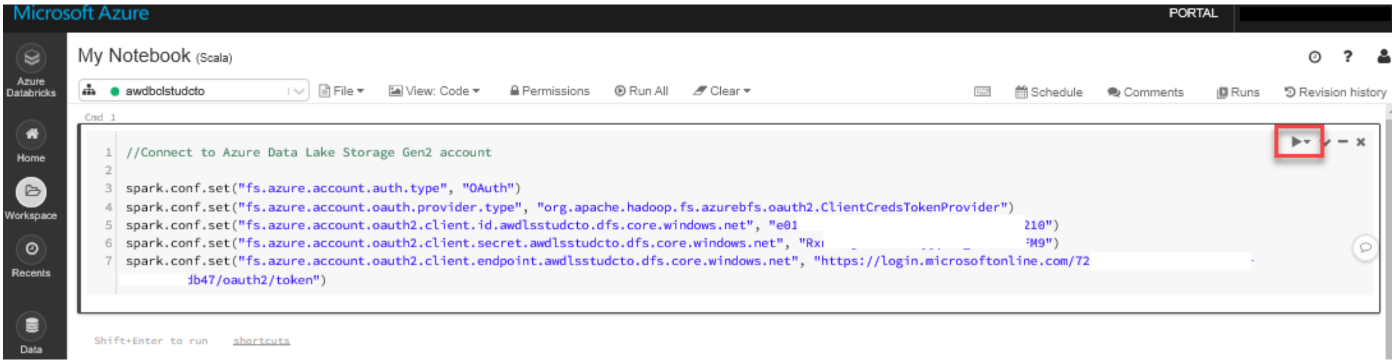
   > **Note**: This will open up a Notebook with the title My Notebook (Scala).

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.ClientCredsTokenProvider")
   spark.conf.set("fs.azure.account.oauth2.client.id.<storage-account-name>.dfs.core.windows.net", "<application-id>")
   spark.conf.set("fs.azure.account.oauth2.client.secret.<storage-account-name>.dfs.core.windows.net", "<authentication-key>")
   spark.conf.set("fs.azure.account.oauth2.client.endpoint.<storage-account-name>.dfs.core.windows.net", "https://login.microsof
   ```

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.



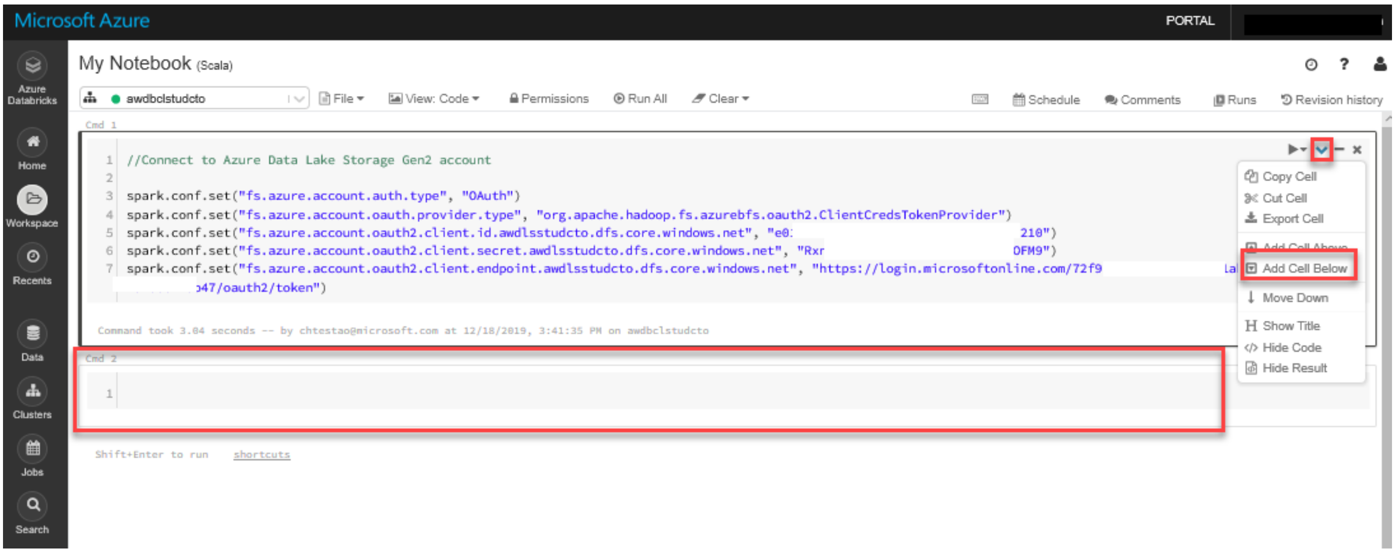
```
//Connect to Azure Data Lake Storage Gen2 account

spark.conf.set("fs.azure.account.auth.type", "OAuth")
spark.conf.set("fs.azure.account.oauth.provider.type", "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider")
spark.conf.set("fs.azure.account.oauth2.client.id.awdlsstudcto.dfs.core.windows.net", "e01[redacted]210")
spark.conf.set("fs.azure.account.oauth2.client.secret.awdlsstudcto.dfs.core.windows.net", "Rx[redacted]M9")
spark.conf.set("fs.azure.account.oauth2.client.endpoint.awdlsstudcto.dfs.core.windows.net", "https://login.microsoftonline.com/72[redacted]db47/oauth2/token")
```

> **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"

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**.



```
//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.ClientCredsTokenProvider")
spark.conf.set("fs.azure.account.oauth2.client.id.awdlsstudcto.dfs.core.windows.net", "e0[redacted]210")
spark.conf.set("fs.azure.account.oauth2.client.secret.awdlsstudcto.dfs.core.windows.net", "Rxr[redacted]OFM9")
spark.conf.set("fs.azure.account.oauth2.client.endpoint.awdlsstudcto.dfs.core.windows.net", "https://login.microsoftonline.com/72f9[redacted]47/oauth2/token")

Command took 3.04 seconds -- by chtestao@microsoft.com at 12/18/2019, 3:41:35 PM on awdbclstudcto
```

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/preferences.json")
```

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**.

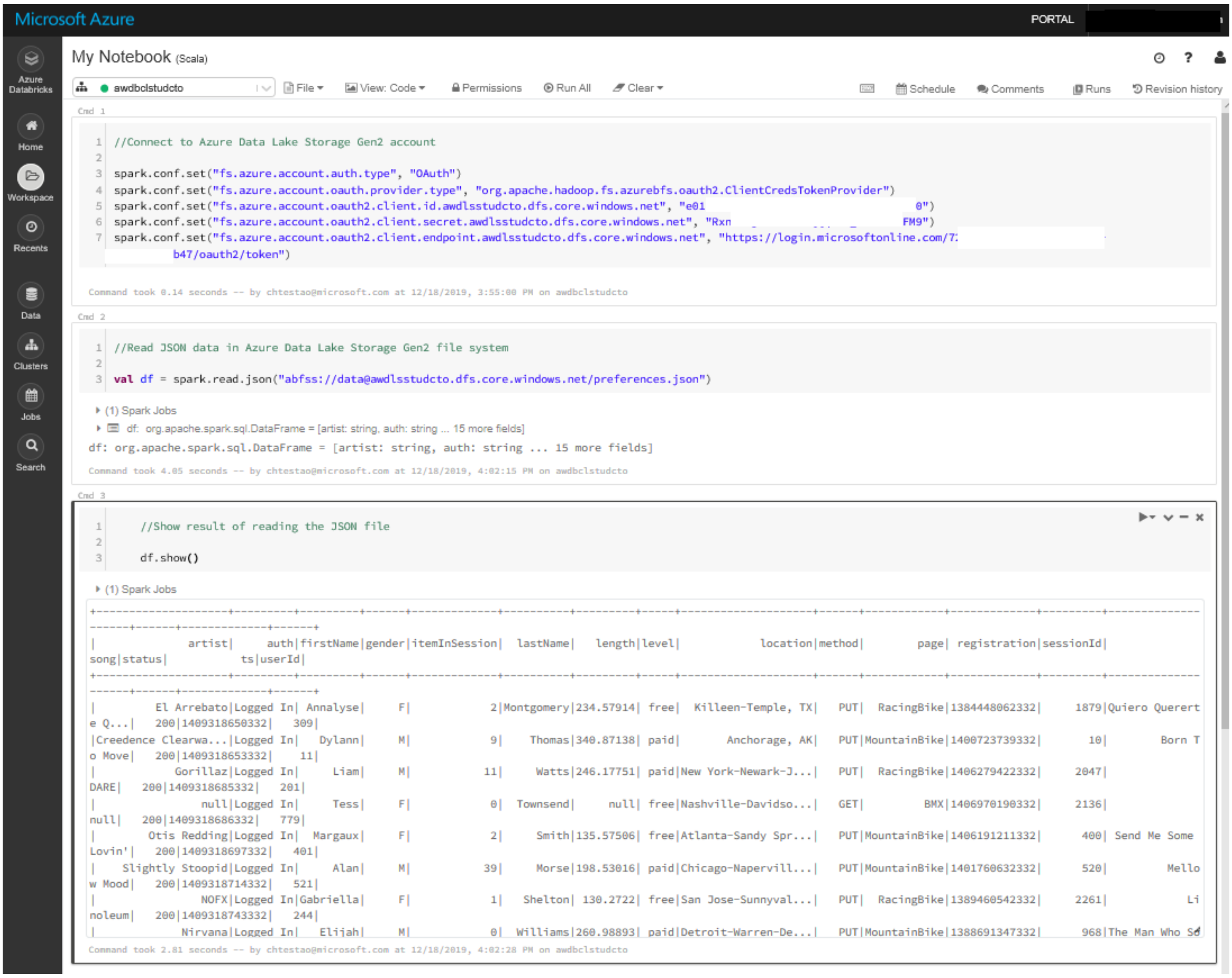
> **Note** A message will be returned at the bottom of the cell that states that a Spark job has executed and "Command took 0.0X seconds -- by person at 4/4/2019, 2:46:48 PM on awdbclstudxx"

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()
```



[redacted]

```
//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.ClientCredsTokenProvider")
spark.conf.set("fs.azure.account.oauth2.client.id.awdlsstudcto.dfs.core.windows.net", "e01                                   0")
spark.conf.set("fs.azure.account.oauth2.client.secret.awdlsstudcto.dfs.core.windows.net", "Rxn                              FM9")
spark.conf.set("fs.azure.account.oauth2.client.endpoint.awdlsstudcto.dfs.core.windows.net", "https://login.microsoftonline.com/7:
          b47/oauth2/token")
```

Command took 0.14 seconds -- by chtestao@microsoft.com at 12/18/2019, 3:55:00 PM on awdbclstudcto

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

val df = spark.read.json("abfss://data@awdlsstudcto.dfs.core.windows.net/preferences.json")
```

▸ (1) Spark Jobs
▸ df: org.apache.spark.sql.DataFrame = [artist: string, auth: string ... 15 more fields]

```
df: org.apache.spark.sql.DataFrame = [artist: string, auth: string ... 15 more fields]
```

Command took 4.05 seconds -- by chtestao@microsoft.com at 12/18/2019, 4:02:15 PM on awdbclstudcto

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

    df.show()
```

▸ (1) Spark Jobs

```
+--------------------+----------+---------+------+-------------+----------+---------+-----+--------------------+------+-------------+-------------+---------+--------------
------+------+-------------+------+
|              artist|      auth|firstName|gender|itemInSession|  lastName|   length|level|            location|method|         page| registration|sessionId|
song|status|           ts|userId|
+--------------------+----------+---------+------+-------------+----------+---------+-----+--------------------+------+-------------+-------------+---------+--------------
------+------+-------------+------+
|         El Arrebato|Logged In| Annalyse|     F|            2|Montgomery|234.57914| free|  Killeen-Temple, TX|   PUT|   RacingBike|1384448062332|     1879|Quiero Querert
e Q...|   200|1409318650332|   309|
|Creedence Clearwa...|Logged In|   Dylann|     M|            9|    Thomas|340.87138| paid|       Anchorage, AK|   PUT|MountainBike|1400723739332|       10|        Born T
o Move|   200|1409318653332|    11|
|             Gorillaz|Logged In|     Liam|     M|           11|     Watts|246.17751| paid|New York-Newark-J...|   PUT|   RacingBike|1406279422332|     2047|
DARE|   200|1409318685332|   201|
|                 null|Logged In|     Tess|     F|            0|  Townsend|     null| free|Nashville-Davidso...|   GET|          BMX|1406970190332|     2136|
null|   200|1409318686332|   779|
|         Otis Redding|Logged In|  Margaux|     F|            2|     Smith|135.57506| free|Atlanta-Sandy Spr...|   PUT|MountainBike|1406191211332|      400| Send Me Some
Lovin'|   200|1409318697332|   401|
|     Slightly Stoopid|Logged In|     Alan|     M|           39|     Morse|198.53016| paid|Chicago-Napervill...|   PUT|MountainBike|1401760632332|      520|         Mello
w Mood|   200|1409318714332|   521|
|                 NOFX|Logged In|Gabriella|     F|            1|   Shelton| 130.2722| free|San Jose-Sunnyval...|   PUT|   RacingBike|1389460542332|     2261|            Li
noleum|   200|1409318743332|   244|
|              Nirvana|Logged In|   Elijah|     M|            0|  Williams|260.98893| paid|Detroit-Warren-De...|   PUT|MountainBike|1388691347332|      968|The Man Who Sо
```

Command took 2.81 seconds -- by chtestao@microsoft.com at 12/18/2019, 4:02:28 PM on awdbclstudcto

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.

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

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**.

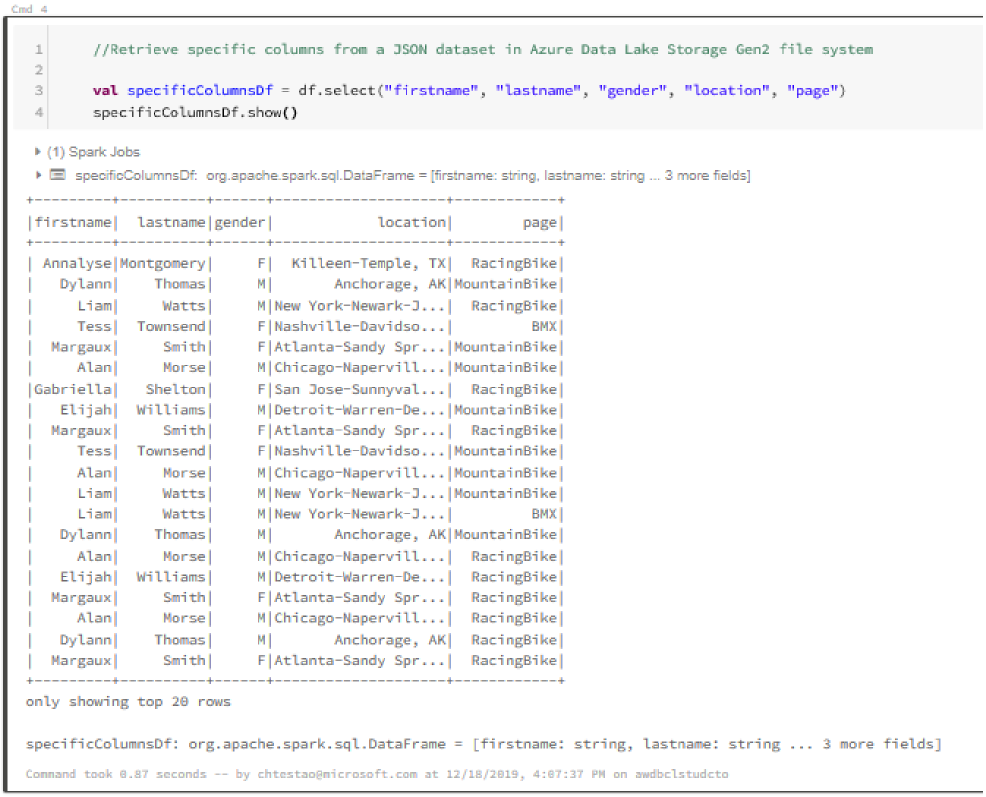
2. In the Notebook, in the cell **Cmd 4**, copy the following code and paste it into the cell:

```
//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"



Cmd 4

```
//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()
```

▸ (1) Spark Jobs

▸ specificColumnsDf: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 3 more fields]

```
+---------+----------+------+--------------------+------------+
|firstname|  lastname|gender|            location|        page|
+---------+----------+------+--------------------+------------+
| Annalyse|Montgomery|     F|  Killeen-Temple, TX|  RacingBike|
|   Dylann|    Thomas|     M|       Anchorage, AK|MountainBike|
|     Liam|     Watts|     M|New York-Newark-J...|  RacingBike|
|     Tess|  Townsend|     F|Nashville-Davidso...|         BMX|
|  Margaux|     Smith|     F|Atlanta-Sandy Spr...|MountainBike|
|     Alan|     Morse|     M|Chicago-Napervill...|MountainBike|
|Gabriella|   Shelton|     F|San Jose-Sunnyval...|  RacingBike|
|   Elijah|  Williams|     M|Detroit-Warren-De...|MountainBike|
|  Margaux|     Smith|     F|Atlanta-Sandy Spr...|  RacingBike|
|     Tess|  Townsend|     F|Nashville-Davidso...|MountainBike|
|     Alan|     Morse|     M|Chicago-Napervill...|MountainBike|
|     Liam|     Watts|     M|New York-Newark-J...|MountainBike|
|     Liam|     Watts|     M|New York-Newark-J...|         BMX|
|   Dylann|    Thomas|     M|       Anchorage, AK|MountainBike|
|     Alan|     Morse|     M|Chicago-Napervill...|  RacingBike|
|   Elijah|  Williams|     M|Detroit-Warren-De...|  RacingBike|
|  Margaux|     Smith|     F|Atlanta-Sandy Spr...|  RacingBike|
|     Alan|     Morse|     M|Chicago-Napervill...|  RacingBike|
|   Dylann|    Thomas|     M|       Anchorage, AK|  RacingBike|
|  Margaux|     Smith|     F|Atlanta-Sandy Spr...|  RacingBike|
+---------+----------+------+--------------------+------------+
only showing top 20 rows

specificColumnsDf: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 3 more fields]
```

Command took 0.87 seconds -- by chtestao@microsoft.com at 12/18/2019, 4:07:37 PM on awdbclstudcto

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**.

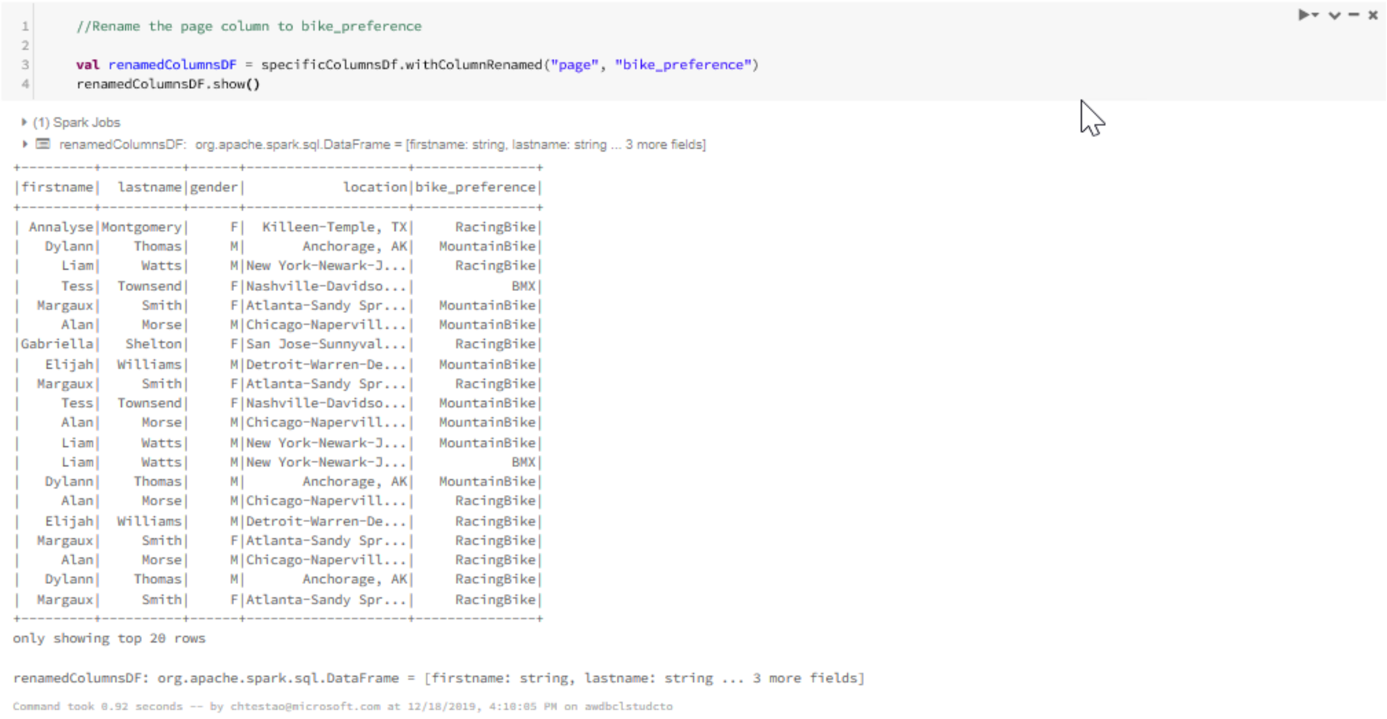
2. In the Notebook, in the cell **Cmd 5**, copy the following code and paste it into the cell:

```
//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"



```
//Rename the page column to bike_preference

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

▸ (1) Spark Jobs
▸ renamedColumnsDF: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 3 more fields]

```
+---------+----------+------+--------------------+---------------+
|firstname|  lastname|gender|            location|bike_preference|
+---------+----------+------+--------------------+---------------+
| Annalyse|Montgomery|     F|  Killeen-Temple, TX|     RacingBike|
|   Dylann|    Thomas|     M|       Anchorage, AK|   MountainBike|
|     Liam|     Watts|     M|New York-Newark-J...|     RacingBike|
|     Tess|  Townsend|     F|Nashville-Davidso...|            BMX|
|  Margaux|     Smith|     F|Atlanta-Sandy Spr...|   MountainBike|
|     Alan|     Morse|     M|Chicago-Napervill...|   MountainBike|
|Gabriella|   Shelton|     F|San Jose-Sunnyval...|     RacingBike|
|   Elijah|  Williams|     M|Detroit-Warren-De...|   MountainBike|
|  Margaux|     Smith|     F|Atlanta-Sandy Spr...|     RacingBike|
|     Tess|  Townsend|     F|Nashville-Davidso...|   MountainBike|
|     Alan|     Morse|     M|Chicago-Napervill...|   MountainBike|
|     Liam|     Watts|     M|New York-Newark-J...|   MountainBike|
|     Liam|     Watts|     M|New York-Newark-J...|            BMX|
|   Dylann|    Thomas|     M|       Anchorage, AK|   MountainBike|
|     Alan|     Morse|     M|Chicago-Napervill...|     RacingBike|
|   Elijah|  Williams|     M|Detroit-Warren-De...|     RacingBike|
|  Margaux|     Smith|     F|Atlanta-Sandy Spr...|     RacingBike|
|     Alan|     Morse|     M|Chicago-Napervill...|     RacingBike|
|   Dylann|    Thomas|     M|       Anchorage, AK|     RacingBike|
|  Margaux|     Smith|     F|Atlanta-Sandy Spr...|     RacingBike|
+---------+----------+------+--------------------+---------------+
only showing top 20 rows

renamedColumnsDF: org.apache.spark.sql.DataFrame = [firstname: string, lastname: string ... 3 more fields]
```

Command took 0.92 seconds -- by chtestao@microsoft.com at 12/18/2019, 4:10:05 PM on awdbclstudcto

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 preferences.json file stored in that data lake. Then a simple query has been created to retrieve data and the column "page" has been renamed to "bike_preference".
   ```

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.

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/master/DBC/05.1-Basic-ETL.dbc?raw=true
```

1. Select **Import**.

2. A folder named **05.1-Basic-ETL** after the import should appear. Select that folder.

3. 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/blob/master/DBC/05.2-
Advanced-ETL.dbc?raw=true
```

1. Select **Import**.

2. A folder named **05.2-Advanced-ETL** after the import should appear. Select that folder.

3. 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.