Microsoft Cloud Workshops

Data Ingestion and Reporting

Walkthrough

April 2019

|  |  |
| --- | --- |
| Authors: | |
| Morgan Price | Data Platform Solution Architect |
| Sanaullah Sanai | Cloud Solution Architect - Analytics & AI |

© 2019 Microsoft Corporation. All rights reserved. This document is confidential and proprietary to Microsoft. Internal use only. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples are for illustration only and are fictitious. No real association is intended or inferred.

Contents

[Overview 1](#_Toc5282267)

[Objectives 1](#_Toc5282268)

[The Workshop 1](#_Toc5282269)

[Delivery 1](#_Toc5282270)

[Duration 2](#_Toc5282271)

[Requirements 2](#_Toc5282272)

[Workshop Step-by-Step 3](#_Toc5282273)

[Lab 01: Setup 3](#_Toc5282274)

[Exercise 1: Access Azure 3](#_Toc5282275)

[Exercise 2: Access GitHub Repository 4](#_Toc5282276)

[Exercise 3: Create Local File Storage Location 5](#_Toc5282277)

[Lab 02: Create Raw Data Repository in Azure 9](#_Toc5282278)

[Exercise 1: Create New Dashboard 9](#_Toc5282279)

[Exercise 2: Create Azure Data Repository 9](#_Toc5282280)

[Exercise 3: Use Azure Storage Explorer (ASE) to Upload Raw Data 15](#_Toc5282281)

[Lab 03: Deploy and Secure the Analytics and Reporting Data Store 20](#_Toc5282282)

[Exercise 1: Deploy an Azure SQL Server and Database Instance 20](#_Toc5282283)

[Exercise 2: Configure and Secure the SQL Database 25](#_Toc5282284)

[Exercise 3: Create the Azure SQL Database Artifacts to Hold WDI Data 30](#_Toc5282285)

[Lab 04: Deploy Azure Data Factory Solution 33](#_Toc5282286)

[Exercise 1: Deploy an Azure Data Factory Instance 33](#_Toc5282287)

[Exercise 2: Create Azure Data Factory Connections 36](#_Toc5282288)

[Exercise 3: Create Azure Data Factory Datasets 42](#_Toc5282289)

[Exercise 4: Create and run the Azure Data Factory Pipeline 53](#_Toc5282290)

[Lab 05: Deploy Azure Databricks for Data Analysis and Prediction 62](#_Toc5282291)

[Exercise 1: TODO: Title 62](#_Toc5282292)

[Lab 06: Develop Power BI Report to Surface Analysis 63](#_Toc5282293)

[Exercise 1: TODO: Title 63](#_Toc5282294)

# Overview

This workshop is designed to guide participants through a common data ingestion and reporting scenario. As we work through this activity, we will explore alternative technologies, architectures and approaches that could be utilized to address different environments and requirements. These include unstructured data, big data, high velocity data and deployment scenarios.

## Objectives

Explore and gain hands on experience with the Azure Data stack of technologies.

Develop a working end to end ingestion, analytics and reporting solution.

Discuss options, architectures and alternative technologies to fit specific requirements.

## The Workshop

The data we will be using is part of the World Development Indicators (WDI) dataset developed by the World Bank. Health data has been extracted from the larger set of datapoints and has been provided in CSV file format in 2 files. One file contains WDI country data and the other file contains yearly indicator records for each country.

We will firstly upload these files into Azure Blob Storage using Azure Storage Explorer. Then we will deploy and secure an Azure SQL Database and create the database artifacts required to store and retrieve the data. Next, we create an Azure Data Factory pipeline that will load our WDI data into the Azure SQL Database artifacts we created earlier. The next step is to connect to this repository using Azure Databricks where we will manipulate a dataset and use it to train a predictive model to provide data insights, we will write these results back to the SQL database. The final step will be to surface these insights and the WDI health data utilizing Power BI in the form of an interactive report.

## Delivery

Our instructors will walk through each exercise and then provide assistance to participants to complete their own implementation.

Please feel free to ask questions of any of us at any time and ask for help when required.

We want to promote an interactive environment and foster 2-way discussions on how and when our technologies and services can be utilized to enable your outcomes, so please feel open to engage and share your ideas.

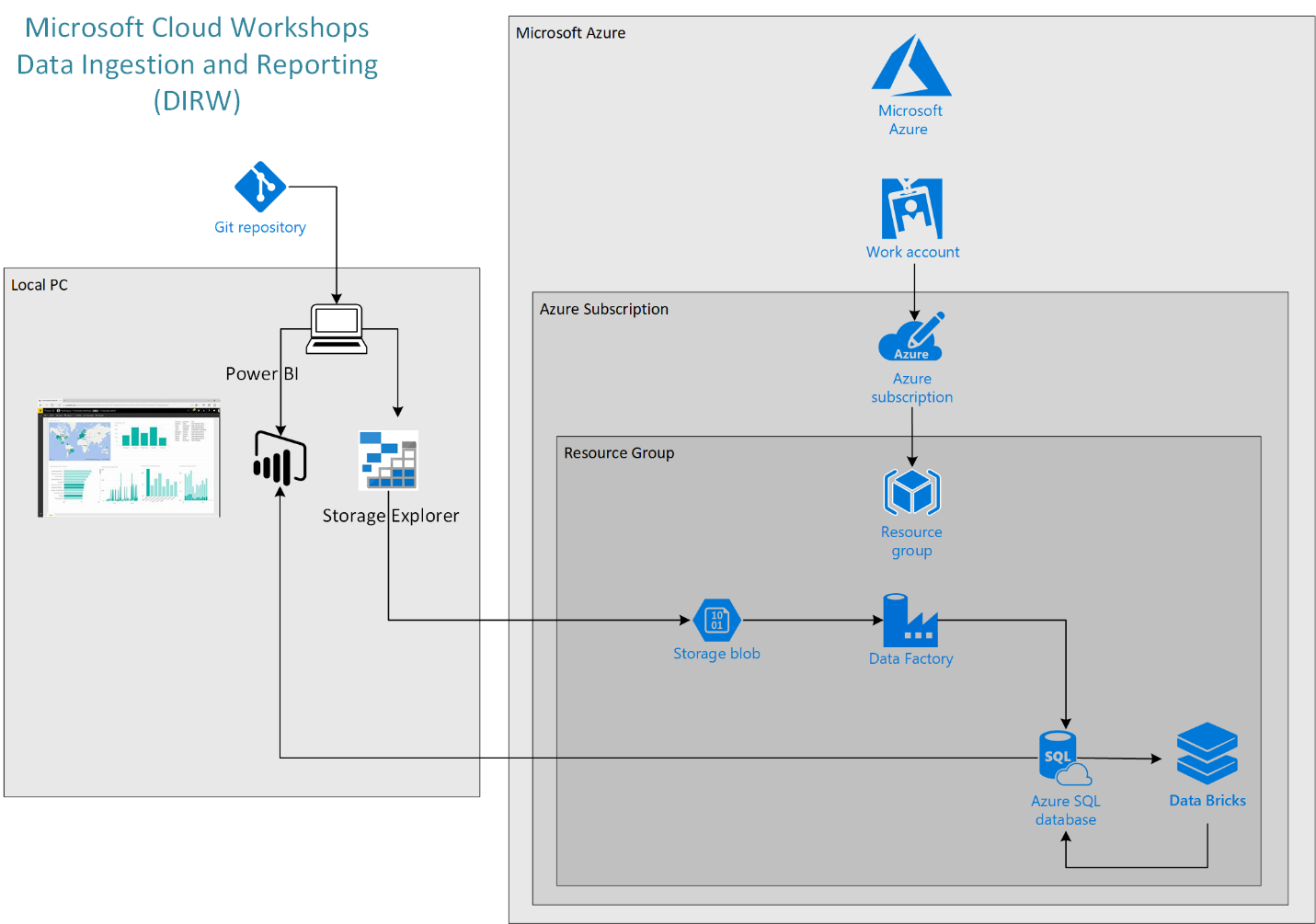
## Duration

4 – 5 Hours.

## Requirements

* Microsoft Azure subscription with permissions to deploy the following services:
  + Azure Blob Storage
  + Azure SQL Server DB
  + Azure Data Factory
  + Azure Databricks
* Laptop with:
  + Azure Storage Explorer installed (latest version: <https://azure.microsoft.com/en-us/features/storage-explorer/>)
  + Power BI Desktop (latest version: <https://powerbi.microsoft.com/en-us/desktop/)>
  + Azure Data Studio installed (latest version: <https://docs.microsoft.com/en-us/sql/azure-data-studio/download?view=sql-server-2017>)

# Workshop Step-by-Step



## Lab 01: Setup

In this lab we will test access to the required online resources for this workshop and create the local raw data repository. In addition we will download required scripts also.

Prerequisites

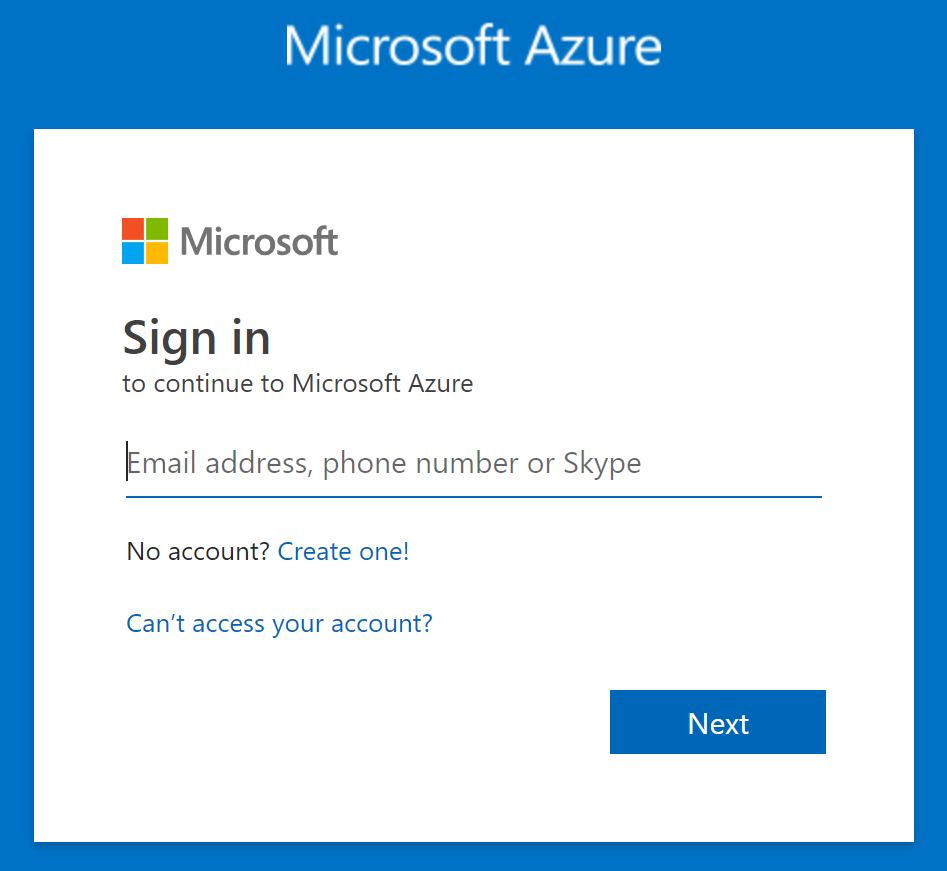
* Microsoft Azure subscription, please have your subscription details available for this workshop. If you do not have your own subscription (recommended), a free trial subscription can be accessed here: <http://azure.microsoft.com/en-us/pricing/free-trial/>

Exercise 1: Access Azure

This is an Azure based workshop, here we will ensure your subscription is good to go.

*Task 1: Access and bookmark the Azure Portal*

1. Open a web browser
2. Navigate to the [Azure portal](https://portal.azure.com/)
3. Enter your login details:



1. You should now see your Azure dashboard, bookmark this page for future reference

Exercise 2: Access GitHub Repository

All files, documentation and other artifacts used in this workshop have been published to a GitHub repository.

*Task 1: Navigate and bookmark GitHub repository*

1. Open a web browser
2. Navigate to <https://github.com/MSFT-MJP/MS_DIRW>



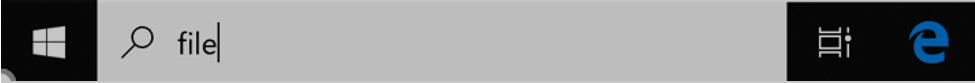
1. Bookmark this repository for future reference

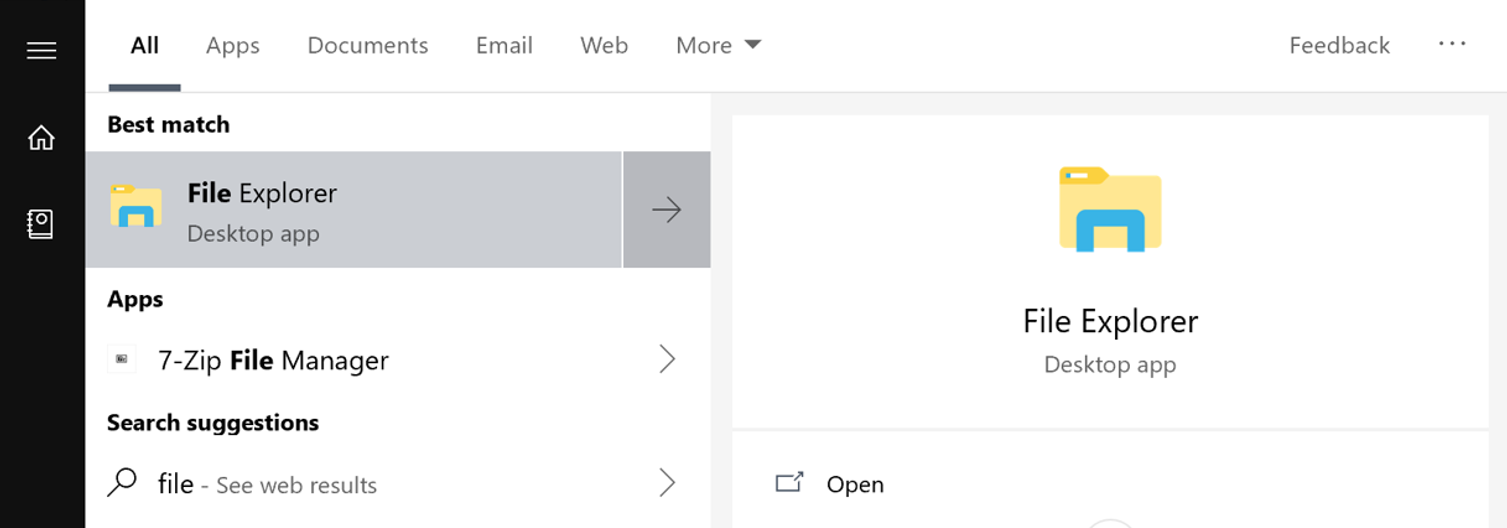
Exercise 3: Create Local File Storage Location

We will need a local copy of the raw data to be used in this workshop.

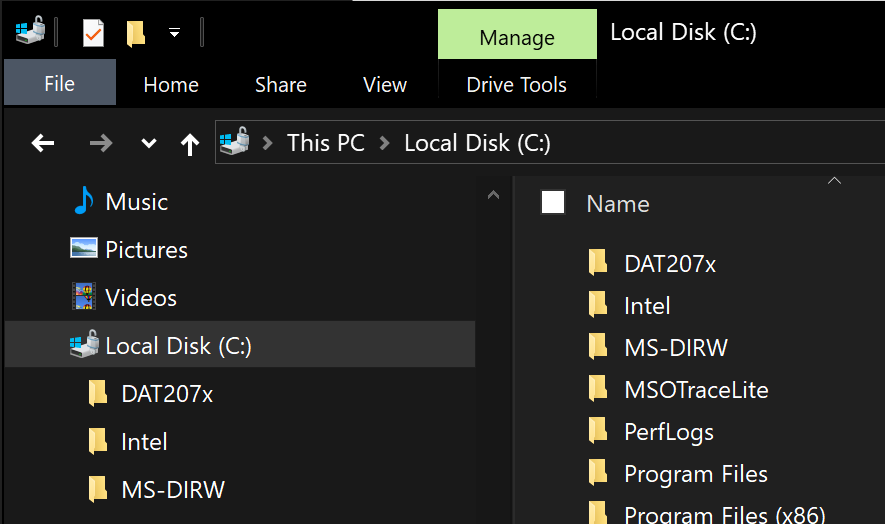
*Task 1: Create local folder*

1. Open File Explorer



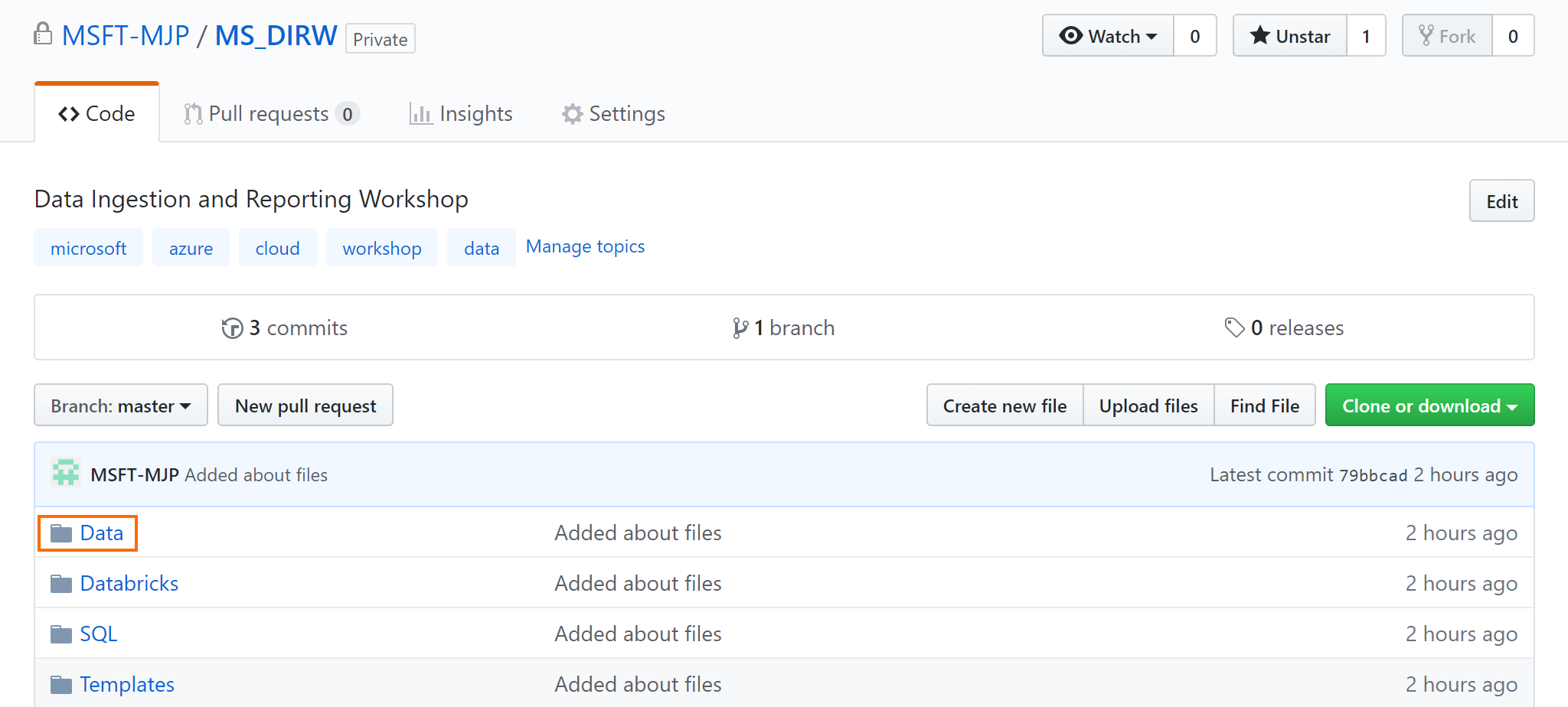


1. Create the following folder: “C:\MS-DIRW”

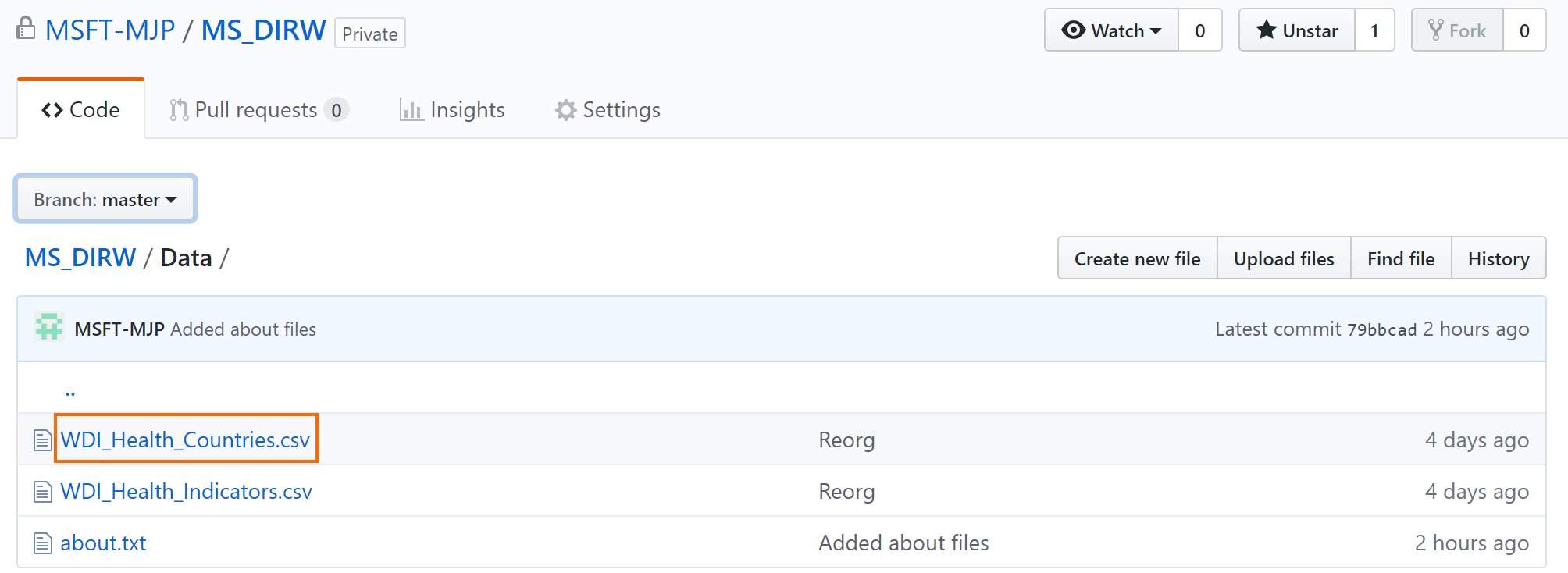


*Task 2: Download Raw Data to the Local Machine*

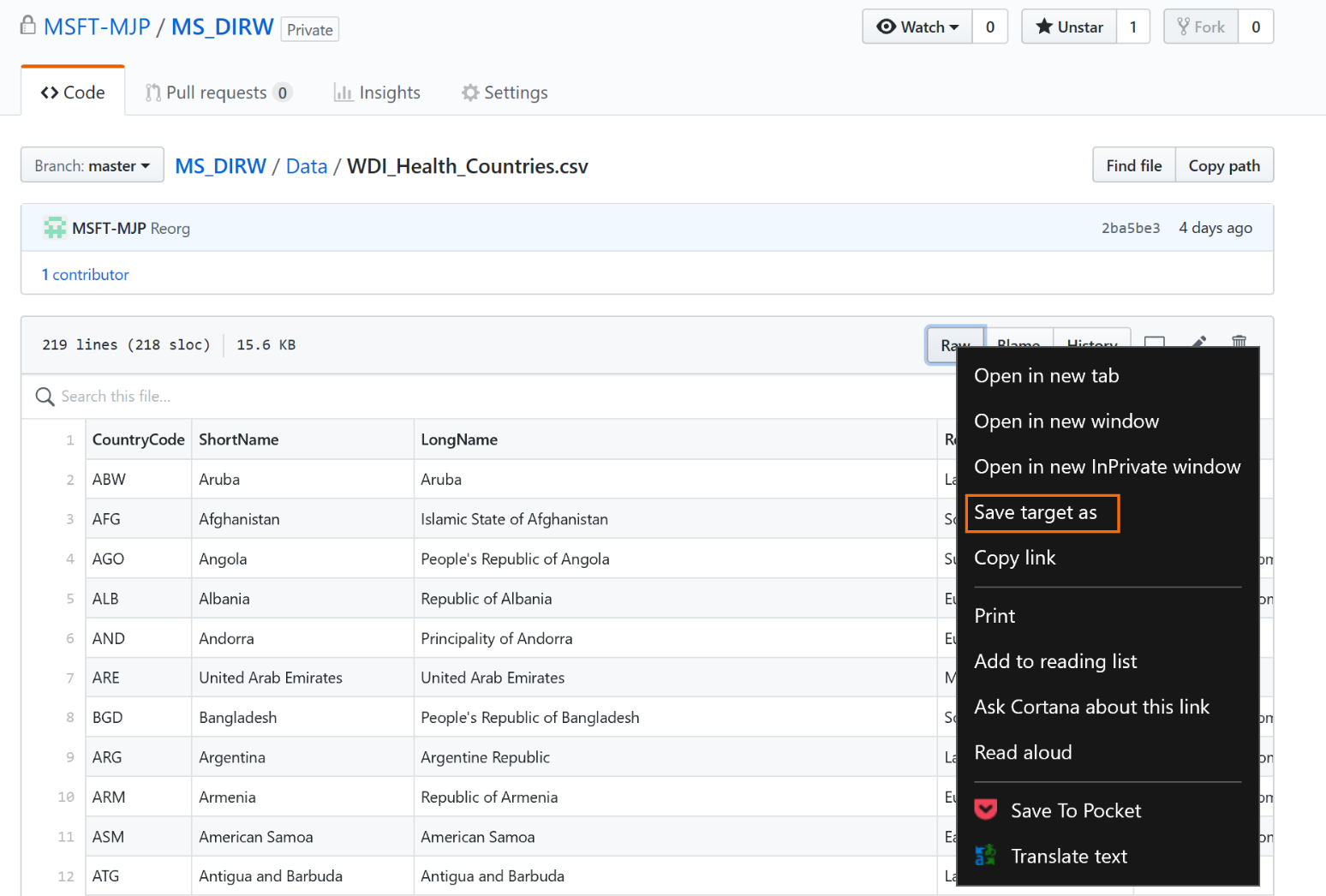
1. Navigate to the [Data Folder](https://github.com/MSFT-MJP/MS_DIRW/tree/master/Data) in the GitHub MS-DIRW repository



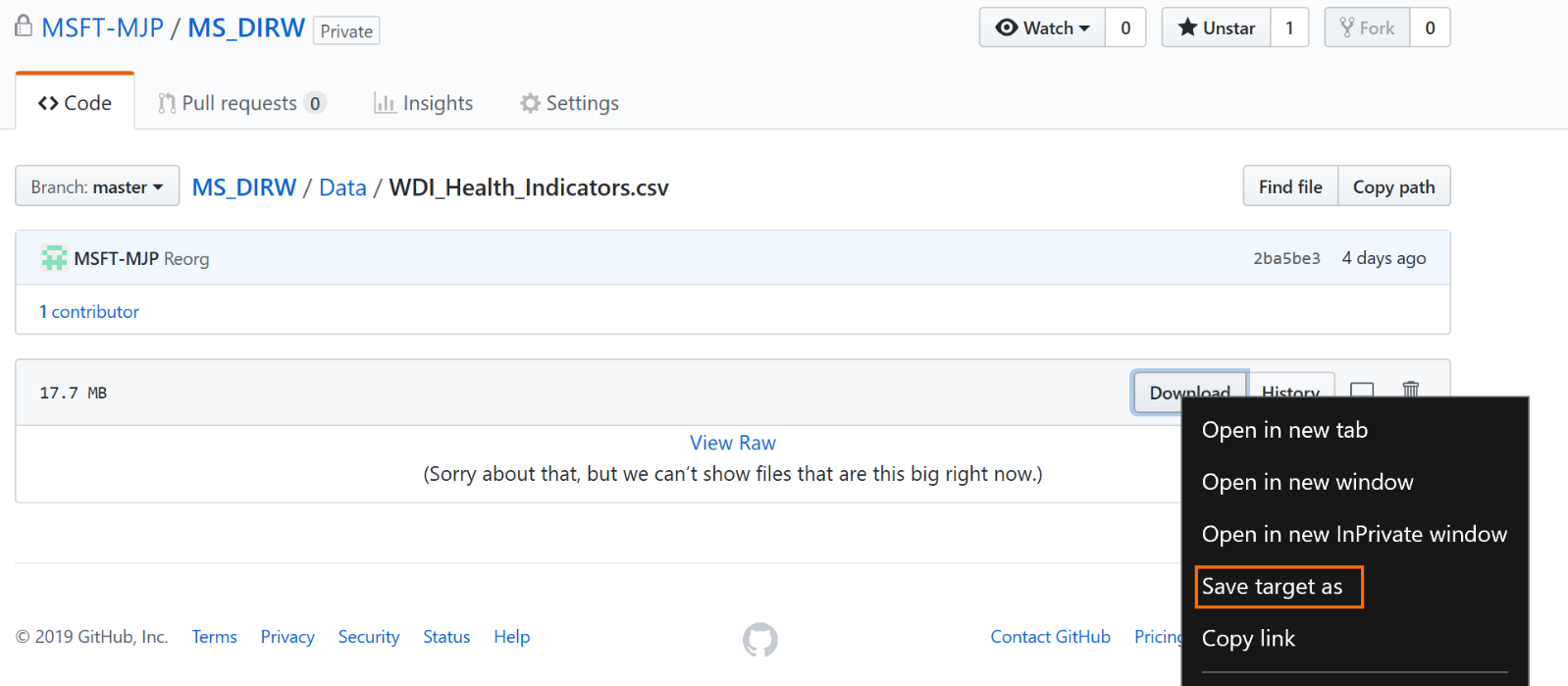
1. Click on [WDI\_Health\_Countries.csv](https://github.com/MSFT-MJP/MS_DIRW/blob/master/Data/WDI_Health_Countries.csv)



1. Right click on the Raw button and “Save target as” (or equivalent depending on your browser) and save the file in the C:\MS-DIRW folder on your local machine.



1. As the [WDI\_Health\_Indicators.csv](https://github.com/MSFT-MJP/MS_DIRW/blob/master/Data/WDI_Health_Countries.csv) file is a larger file the download option is available instead of the view Raw option. Right click on the “Download” button and “Save target as” (or equivalent depending on your browser) and save the file in the C:\MS-DIRW folder on your local machine.



*Task 3: Download SQL notebook to the Local Machine*

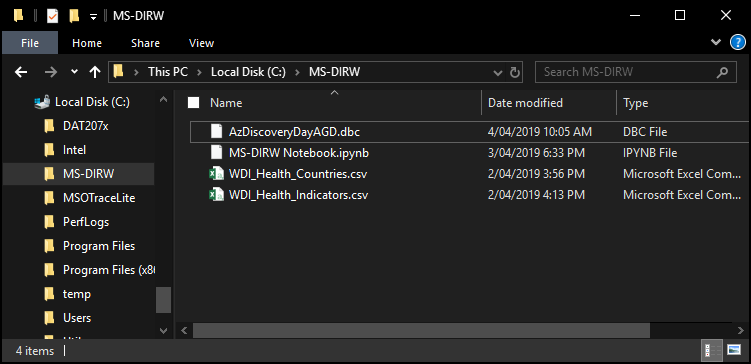
1. Navigate to the [SQL Folder](https://github.com/MSFT-MJP/MS_DIRW/tree/master/SQL) in the GitHub MS-DIRW repository
2. Click on [MS-DIRW Notebook.ipynb](https://github.com/MSFT-MJP/MS_DIRW/blob/master/SQL/MS-DIRW%20Notebook.ipynb)
3. Right click on the Raw button and “Save target as” (or equivalent depending on your browser) and save the file in the C:\MS-DIRW folder on your local machine.
4. Use File Explorer to ensure the following structure exists:

*Task 4: Download Databricks notebook archive to the Local Machine*

1. Navigate to the [Databricks](https://github.com/MSFT-MJP/MS_DIRW/tree/master/Databricks) in the GitHub MS-DIRW repository
2. Click on [AzDiscoveryDayAGD.dbc](https://github.com/MSFT-MJP/MS_DIRW/blob/master/Databricks/AzDiscoveryDayAGD.dbc)
3. Right click on the “Download” button and “Save target as” (or equivalent depending on your browser) and save the file in the C:\MS-DIRW folder on your local machine.

*Task 5: Confirm Local Machine repository files are available*

1. Use File Explorer to ensure the following structure exists:



## Lab 02: Create Raw Data Repository in Azure

In this lab we will create the Azure data repository using Azure Blob Storage and upload our CSV files from our local repository to our Azure repository using Azure Storage Explorer.

Prerequisites

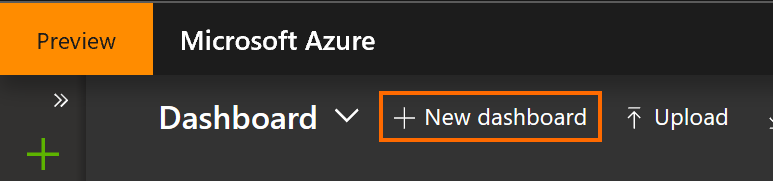
* Previous labs completed successfully
* Microsoft Azure Storage Explorer installed.

Exercise 1: Create New Dashboard

We will create a new dashboard for our workshop to make it easier to access our resources and keep our references all in one place.

*Task 1: Create a new dashboard*

1. Log into the [Azure portal](https://portal.azure.com/)
2. Click on “+ New dashboard”



1. Update the name to “MS-DIRW” and then click the “Done customizing” button

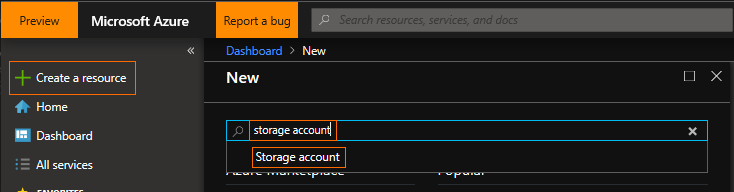


Exercise 2: Create Azure Data Repository

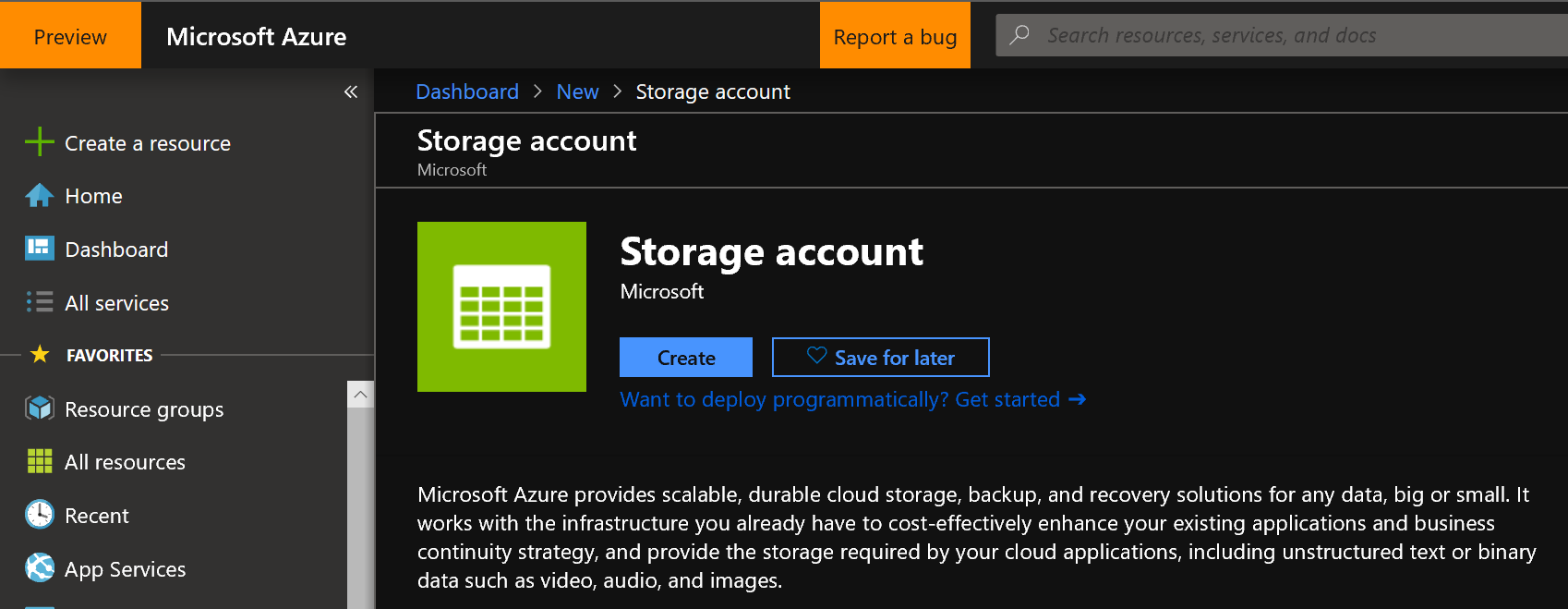
Deploy Azure Blob Storage and create the required structures.

*Task 1: Create a new Storage Account*

1. From your dashboard, click on “Create a resource”, type “Storage Account” and select the “Storage account” entry



1. Click “Create”



1. Fill in the following fields:
   1. Subscription: Select your subscription
   2. Resource group: “Create New”, enter “MS-DIRW” (we will use this resource group for all resources created for this workshop)
   3. Storage account name: Enter “msdirw” and append your initials, e.g. “msdirwmjp”. This name must be globally unique so add additional information if your chosen name has already been used

Remember the name you have chosen so you can reuse this resource group throughout the workshop.

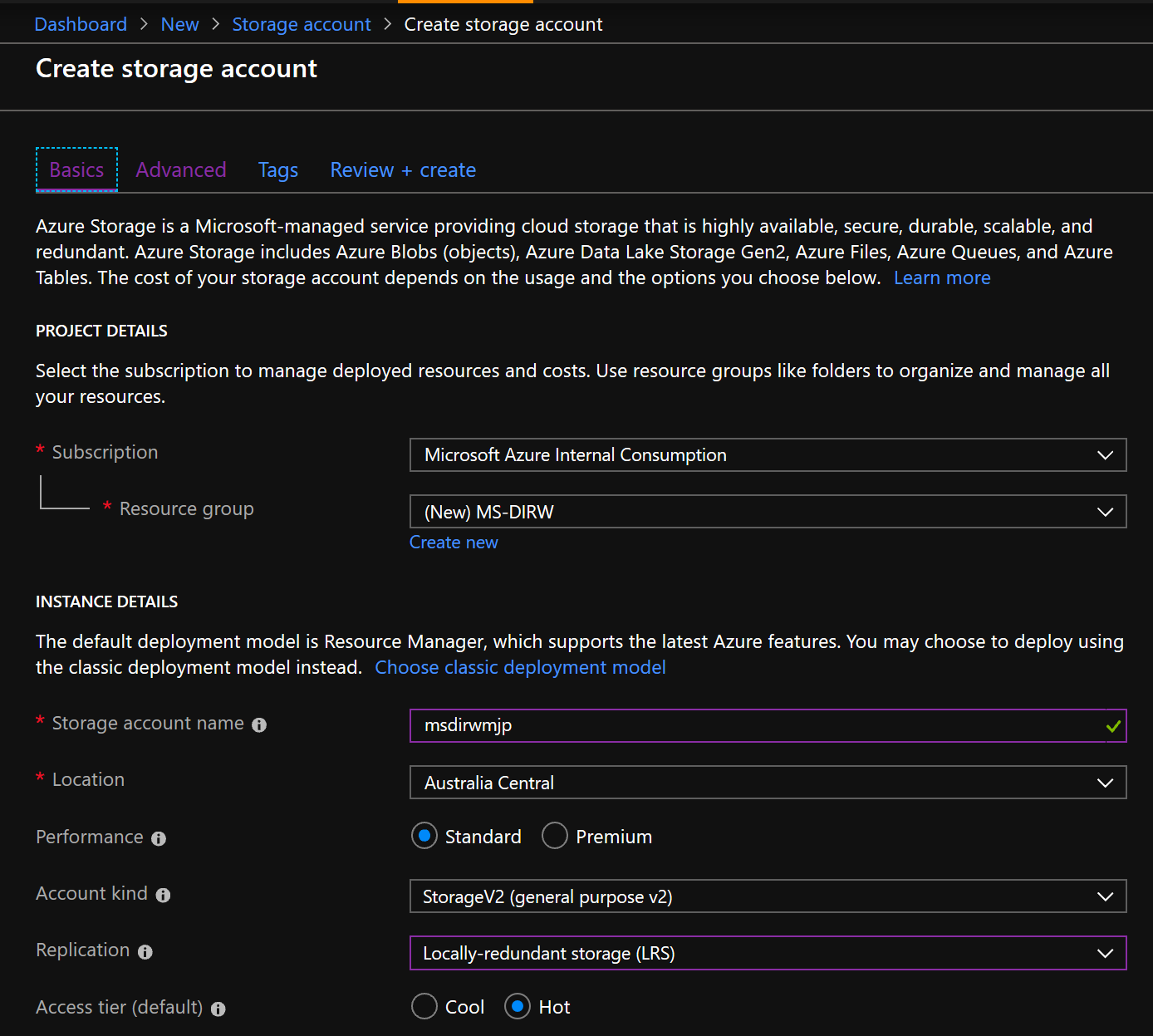
* 1. Location: “Australia Central”

If you use a different location, for performance and cost reasons, use the same location for all resources created for this workshop.

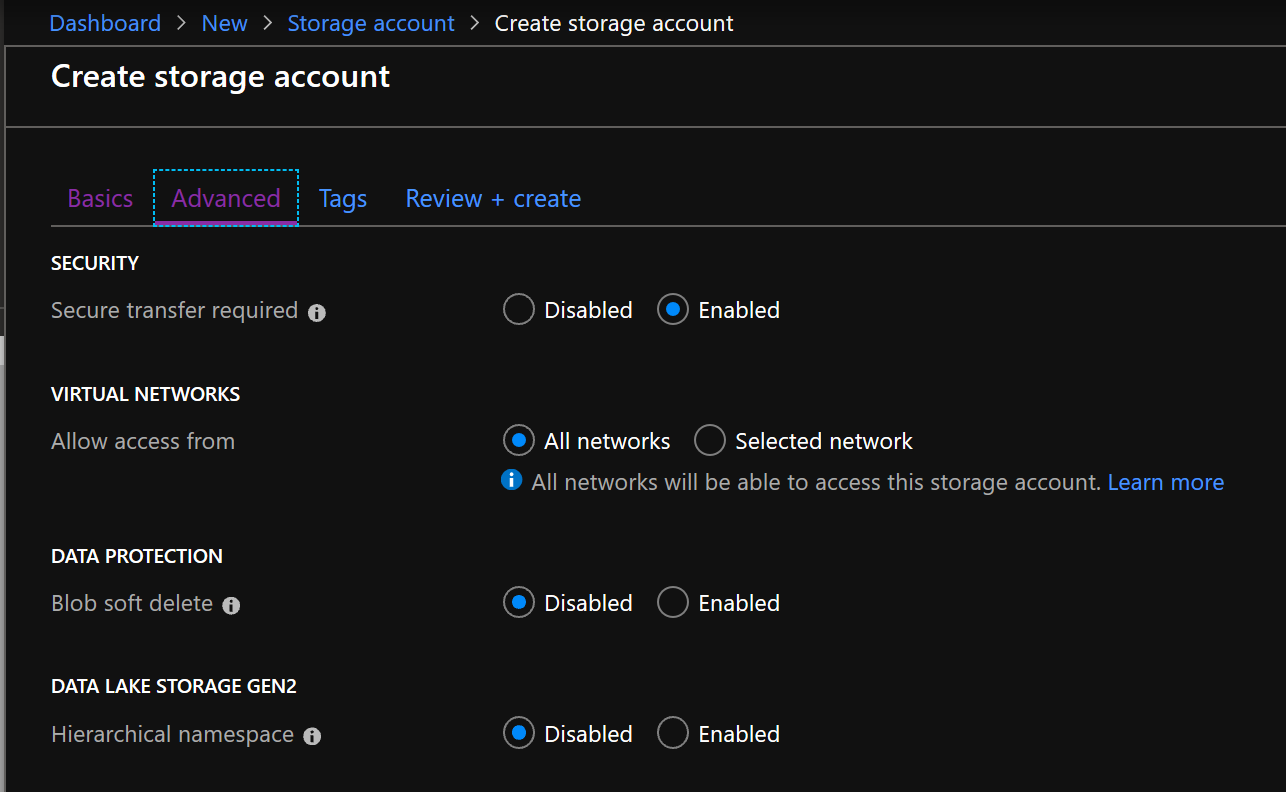
* 1. Performance: Standard, fast enough for this workshop
  2. Account kind: StorageV2 (general purpose v2)
  3. Replication: LRS

We don’t need geo-redundancy for this workshop but for production workloads geo-redundancy is recommended.

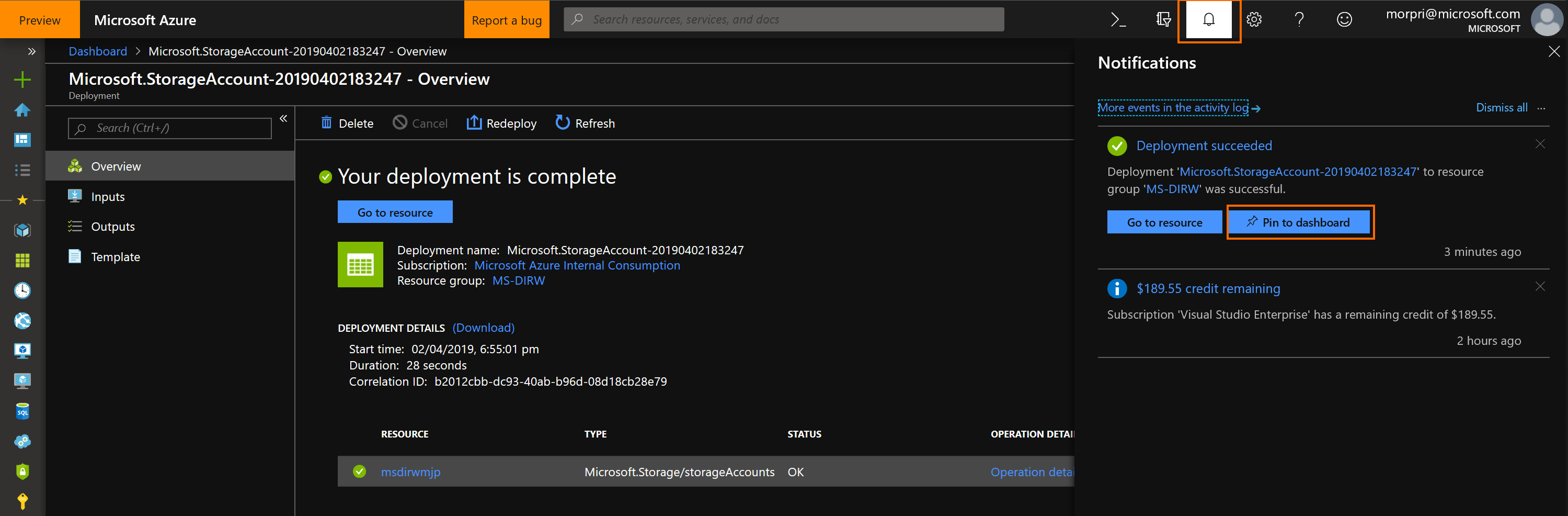
* 1. Access tier: “Hot”



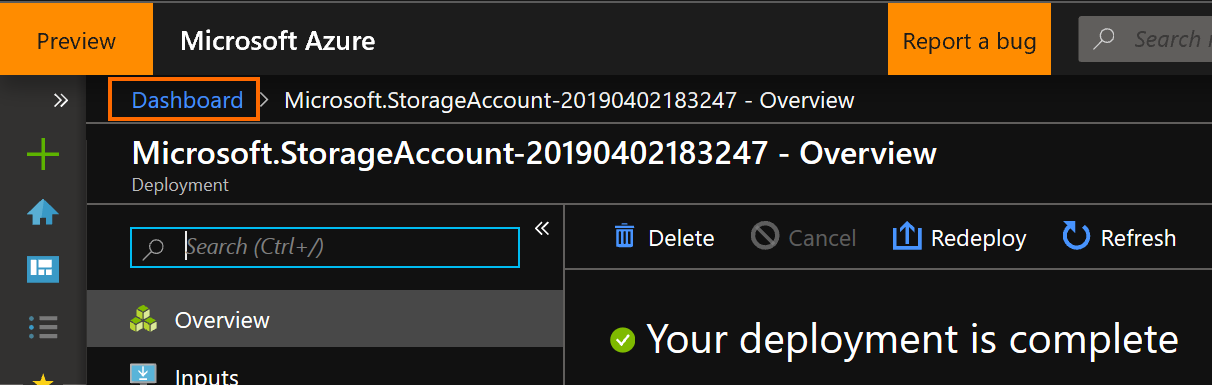
1. Under the “Advanced” tab we will use the defaults, note that we have enabled “Secure transfer” and that we can restrict access to certain networks to reduce surface area



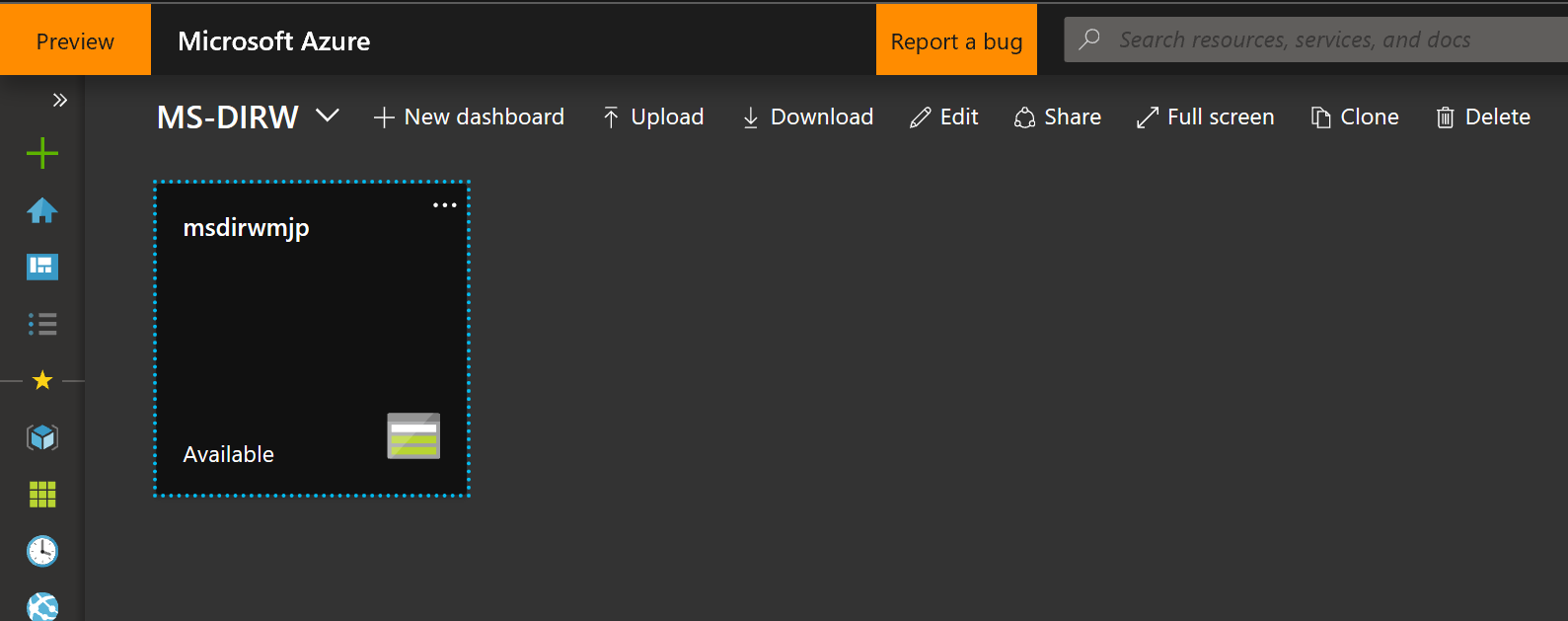
1. Review and Create your Storage account
2. When complete click on the notification (access using the little bell icon at the top right) and then click the “Pin to dashboard” button, this will make it easier to access this resource later



1. Click on the “Dashboard” link in the top left navigation path to get back to the dashboard (you can use this at any time to “return home”)

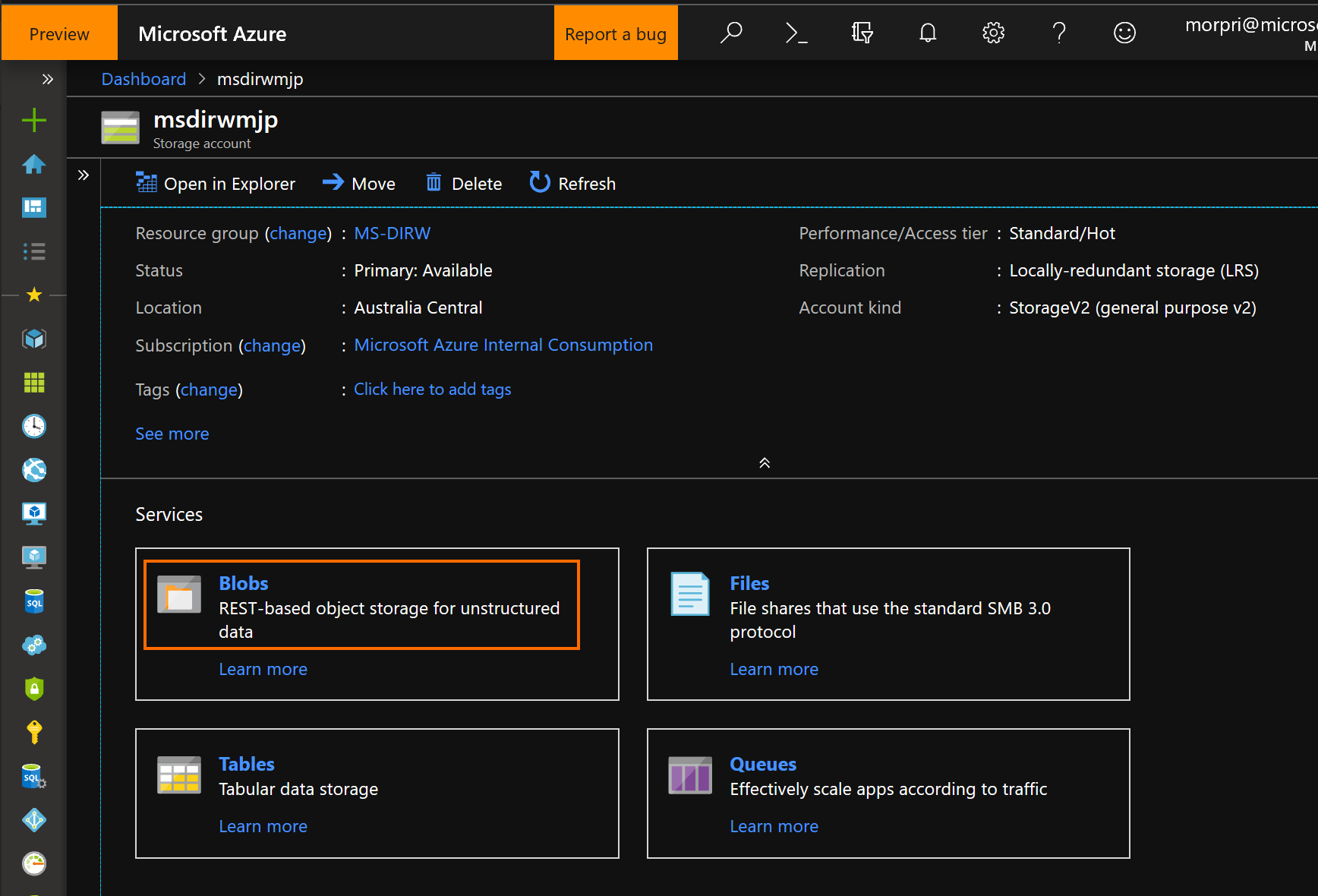


1. You should now see your new Storage account resource on your Dashboard

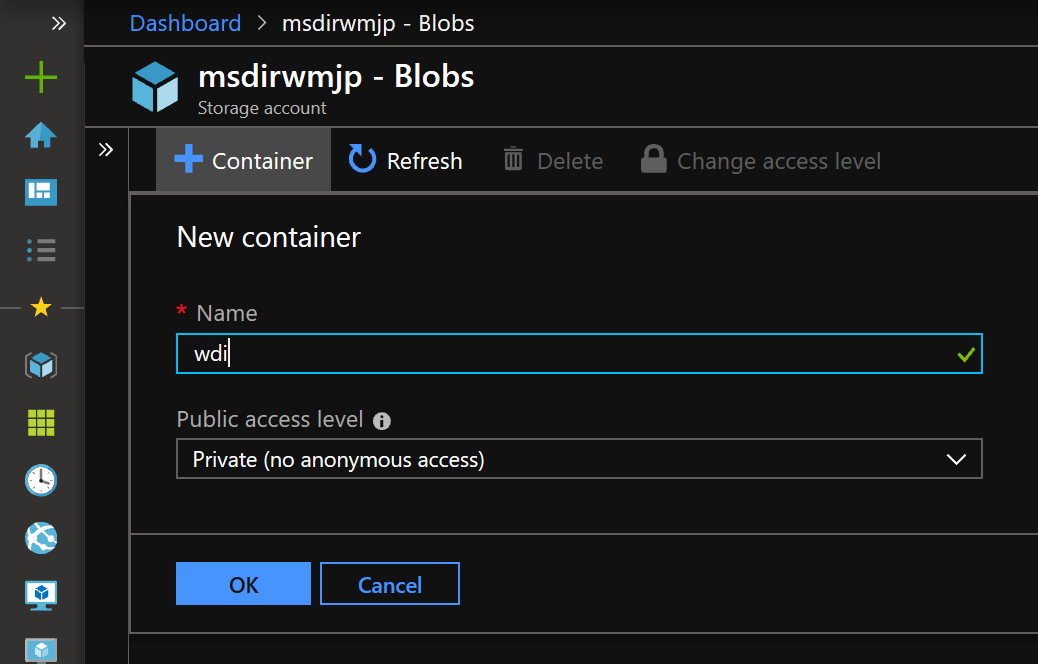


*Task 2: Create a new container to hold our raw data*

1. Click on your Storage account resource
2. Click on Blobs in the Services area



1. Click “+ Container” and name your new container “wdi”, click OK



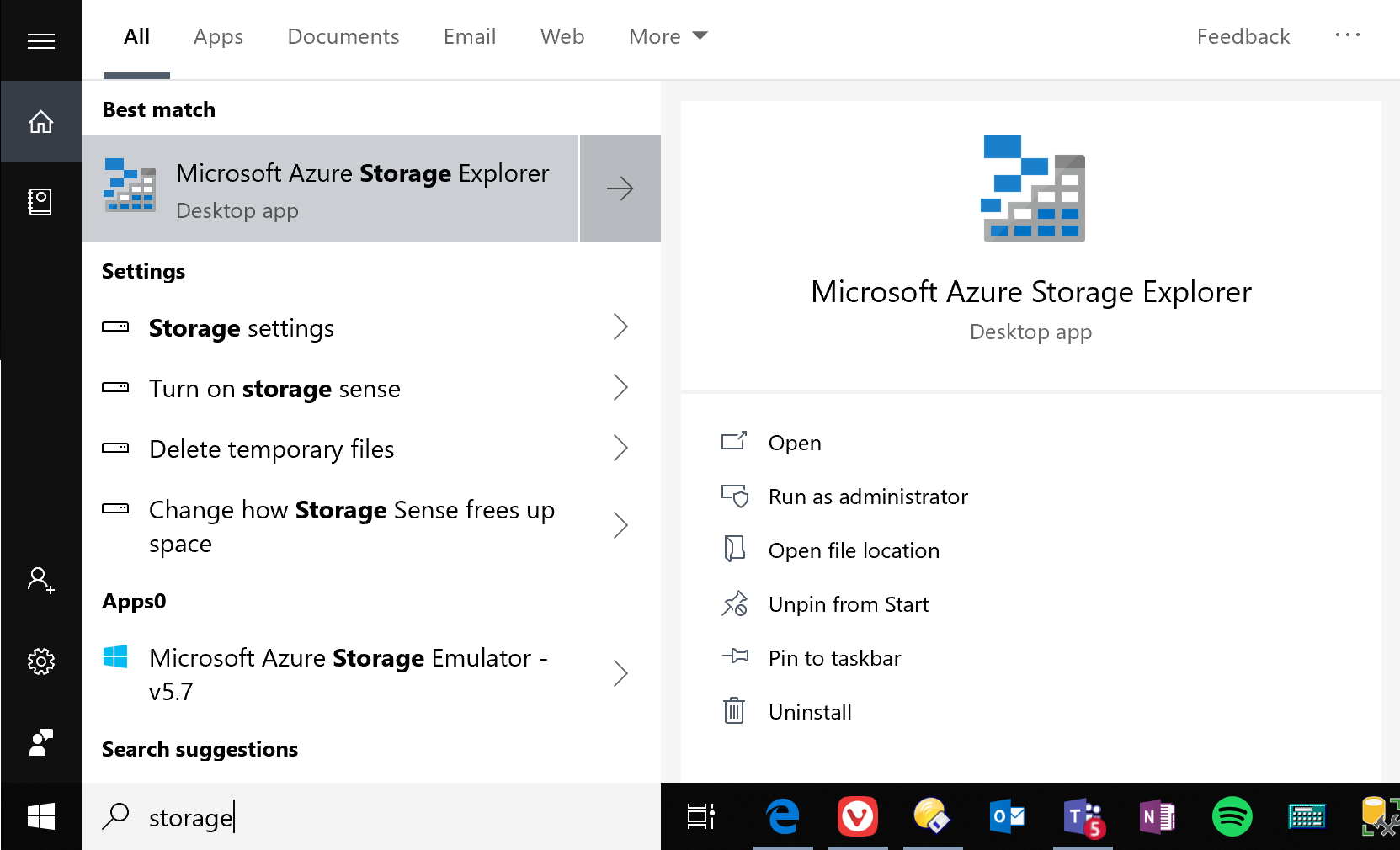
Exercise 3: Use Azure Storage Explorer (ASE) to Upload Raw Data

We will configure and connect ASE to our Azure data repository we have created and upload our raw data CSVs to our wdi container.

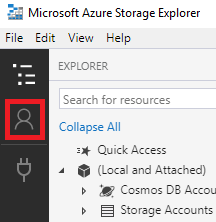
#### Task 1: Connect to an Azure subscription with ASE

The example used in this task is based on connecting a Visual Studio subscription, please substitute this with the subscription you are using for this workshop.

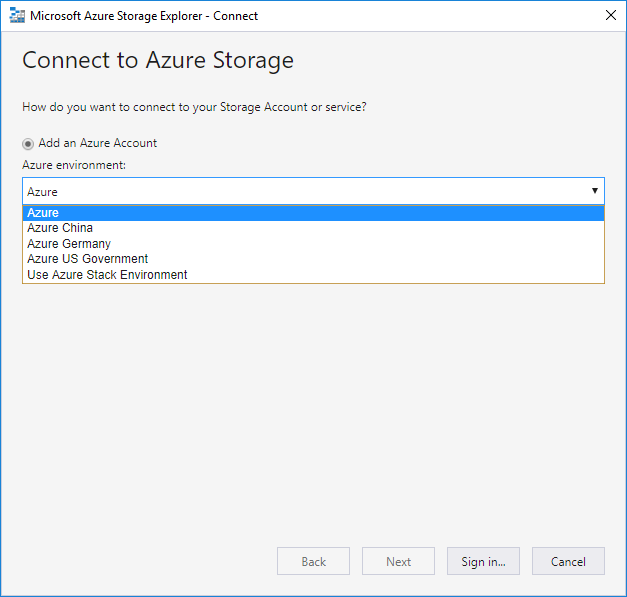
1. Start Storage Explorer



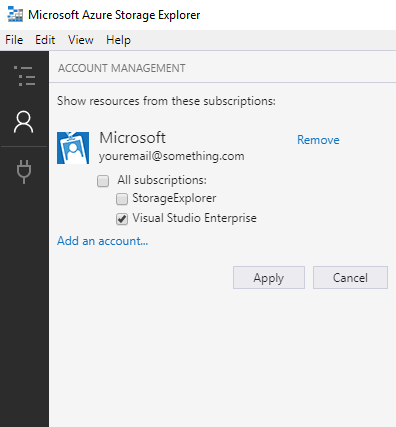
1. In Storage Explorer, select “Manage Accounts” to go to the “Account Management” panel.



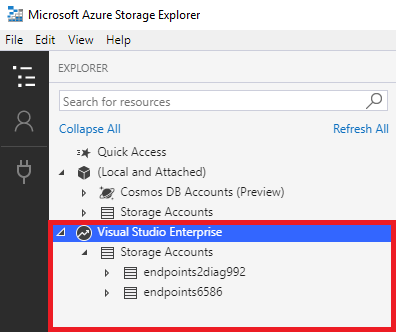
1. The left pane now displays all the Azure accounts you've signed in to. To connect to another account, select “Add an account”
2. Ensure “Azure” is selected in the environment dropdown



1. 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 (Selecting All subscriptions: toggles selecting all or none of the listed Azure subscriptions)

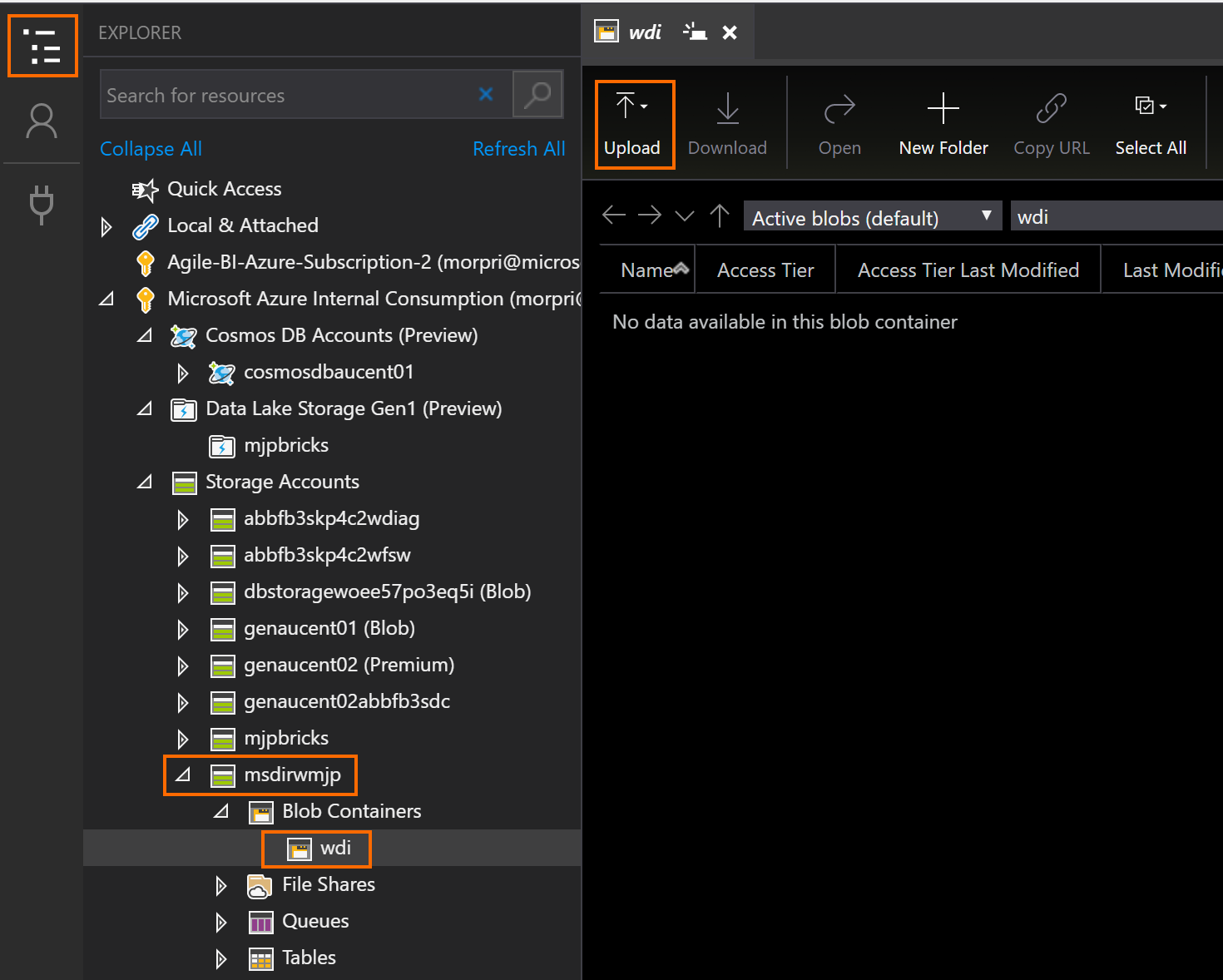


1. The left pane displays the storage accounts associated with the selected Azure subscriptions.

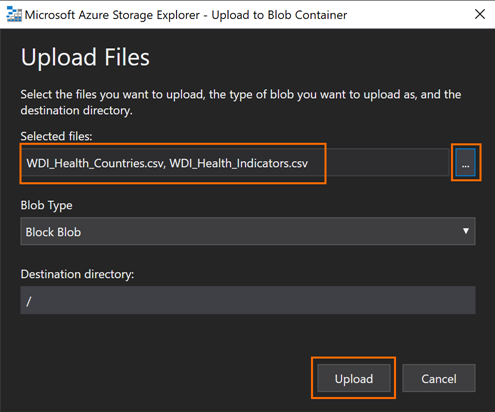


*Task 2: Upload the Raw CSV files*

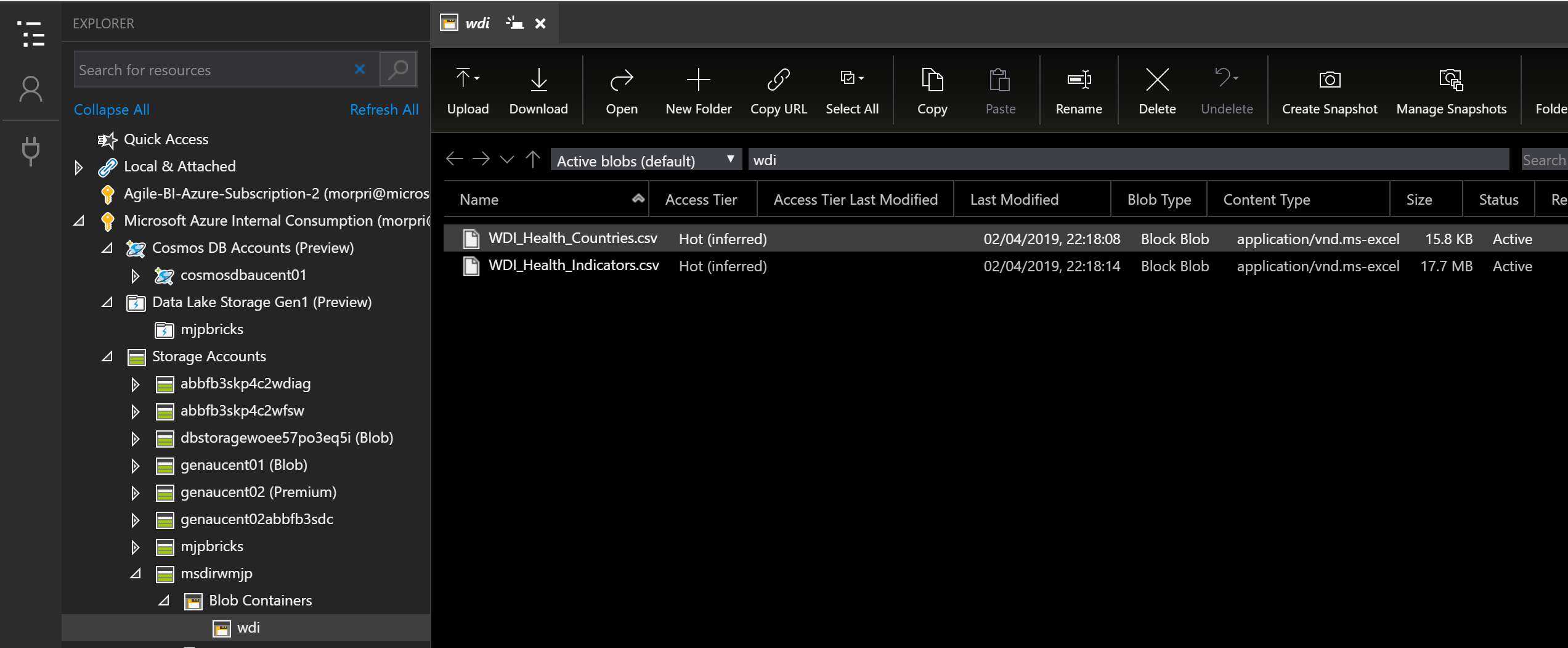
1. In Storage Explorer, navigate to your wdi container you created and select “Upload”



1. Use the “…” button to navigate to C:\MS-DIRW and select our 2 raw data CSV files – WDI\_Health\_Countries.csv and WDI\_Health\_Indicators.csv. Click the Upload button



1. Refresh the view and ensure it looks like this:



## Lab 03: Deploy and Secure the Analytics and Reporting Data Store

In this lab we will deploy and configure an instance of an Azure SQL Database. We will then create the required artifacts (tables, views, users, etc.) to hold a structured version of our data to provide a data platform with the required performance and security for our analytics and reporting requirements.

Prerequisites

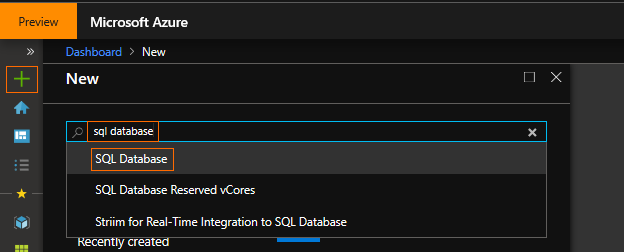
* Previous labs completed successfully
* Azure Data Studio installed

Exercise 1: Deploy an Azure SQL Server and Database Instance

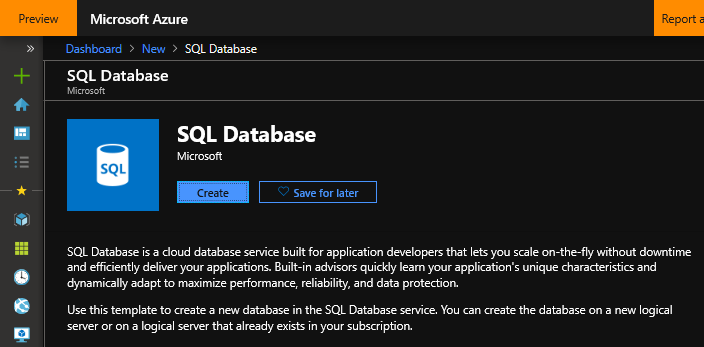
We will create a new server and deploy a database for our WDI data.

*Task 1: Deploy a new Azure SQL Database Instance*

1. Start from your MS-DIRW dashboard.
2. Click on “Create a resource”, type “sql database” in the search box and select the “SQL Database” entry.



1. Click “Create”



1. Fill in the following fields:
   1. Subscription: Select your subscription
   2. Resource group: Select the Resource Group you created when you created your Storage Account in Lab 02. The suggested name was “msdirw” appended with your initials, e.g. “msdirwmjp”. Remember this name had to be globally unique.
   3. Database Name: “MS-DIRW”

Note that we are about to create a new logical server to host our database. A server can host multiple databases so we could use an existing database if we have one and can use this server in in the future to host new database instances.

* 1. Server: Click “Create new” and fill in the following fields:
     1. Server Name: Enter “msdirw” and append your initials, e.g. “msdirwmjp”. This name must be globally unique so add additional information if your chosen name has already been used

Remember the name you have chosen so you can use this name to connect to the server later in the workshop.

* + 1. Server admin login: “msdirwadmin”
    2. Password: “pa$$w0rd”

Of course, outside of this workshop please apply a far more robust and secure password policy. If you wish to use a different password now, please do so.

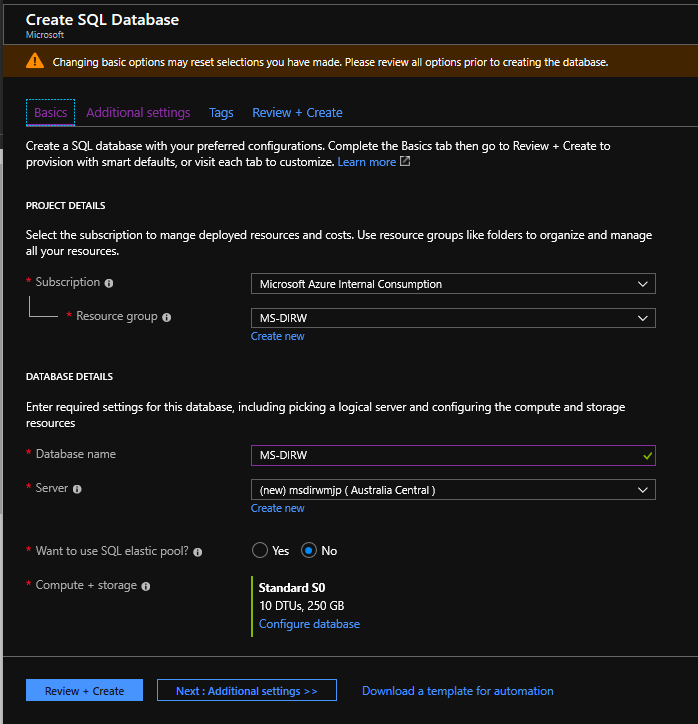
Remember to substitute “pa$$w0rd” with the one you have created whenever you see this password used in this workshop.

* + 1. Location: “Australia Central”

If you are using a different location for other resources deployed for this workshop use the same location for all resources for performance and cost reasons.



* + 1. Click “Select”
  1. Elastic pools: No
  2. Compute + storage: Standard S0, 10 DTUs, 250 GB



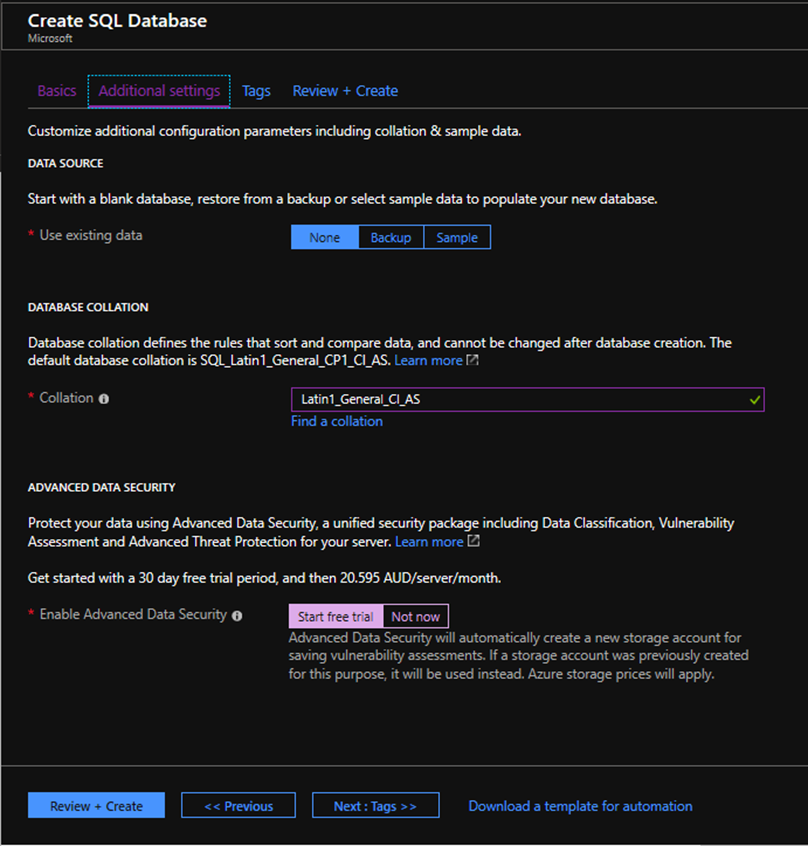
1. On the “Additional settings” tab fill in the following fields:
   1. Use existing data: None
   2. Collation: “Latin1\_General\_CI\_AS”
   3. Enable Advanced Data Security: “Start free trial”

OPTIONAL only, enable this option to explore the advanced security features available with Azure SQL database such as:

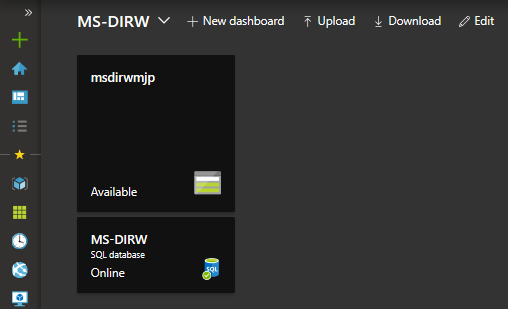
**Advanced Treat Detection** – Get notified of security treats such as suspicious activity & connections, potentially harmful applications, SQL injection and brute force attacks.

**Vulnerability Assessment** - Schedule regular assessments that check for insecure configuration, firewall rules, data classification recommendations, etc.

**Data Classification** – Classify data in your database so appropriate policy can be applied.



1. Click “Review + Create” and then click “Create” to deploy your new Azure SQL Database
2. When complete click on the notification (access using the little bell icon at the top right) and then click the “Pin to dashboard” button, this will make it easier to access this resource later
3. Click on the “Dashboard” link in the top left navigation path to get back to the dashboard
4. You should now see your new SQL Database on your Dashboard

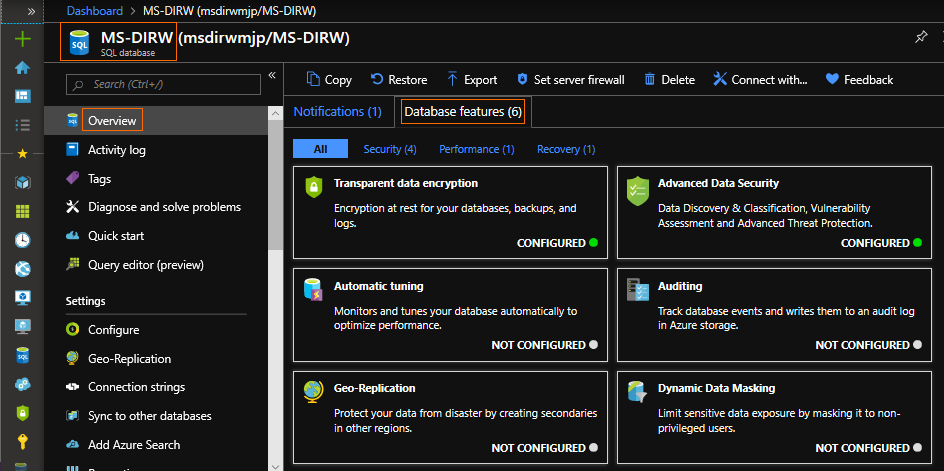


Exercise 2: Configure and Secure the SQL Database

We will ensure our database is secure and automate maintenance tasks.

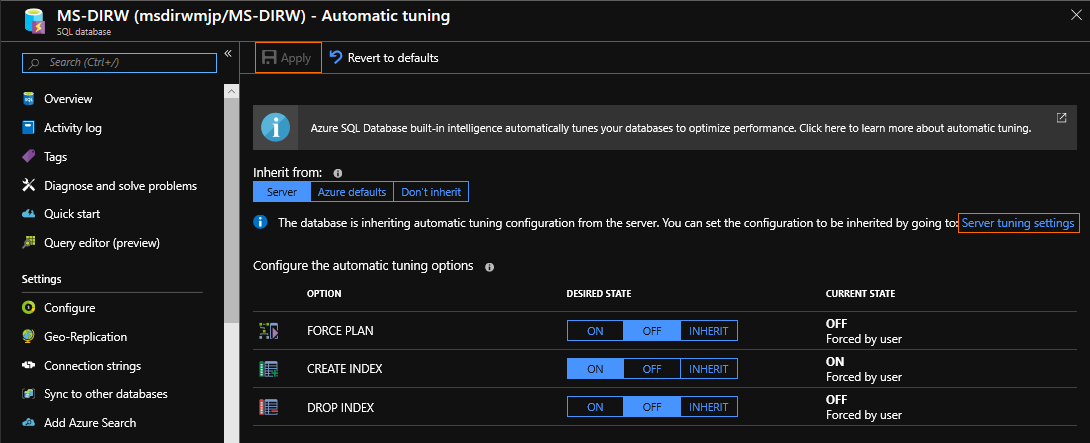
*Task 1: Configure Automatic tuning*

1. From the MS-DIRW dashboard select your SQL database.
2. At the bottom of the database Overview page select “Database features”, then select “Automatic tuning”



1. Configure the options as follows:
   1. FORCE PLAN: OFF, this is used to override join behavior of the query optimizer
   2. CREATE INDEX: ON, this will allow the database to create indexes to speed up data retrieval as it learns from how we access the data
   3. DROP INDEX: OFF, this will allow the database to drop any indexes that are not being used over time to free up space and improve insert and update performance

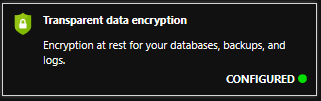
Alternatively, you can click on the “Server tuning settings” link and set these on the server, then set these to “INHERIT” for this database. This will then ensure all new databases created on the server will automatically get these settings as INHERIT is the default setting for new databases.



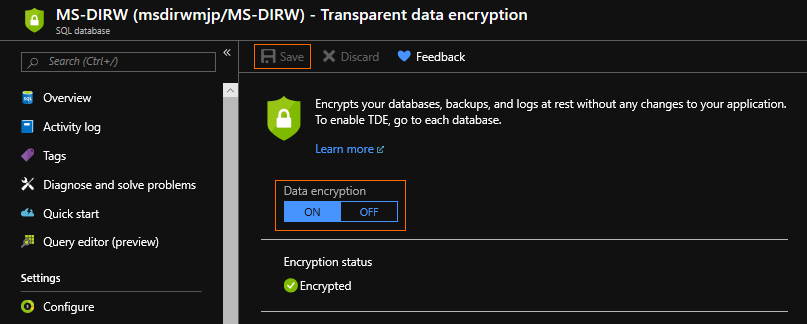
1. Click Apply

*Task 2: Configure Transparent Data Encryption (TDE)*

1. From the SQL database Overview page.
2. At the bottom of the database Overview page “Database features”, check if this database has already had “Transparent data encryption” configured by default, if not select “Transparent data encryption”, otherwise go to the next task. TDE will ensure that all data at rest, including backups and logs are encrypted.

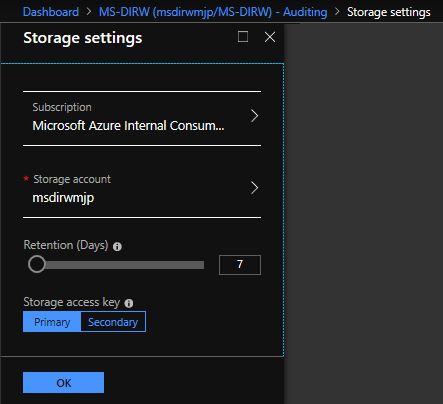


1. Select “ON” and then “Save”

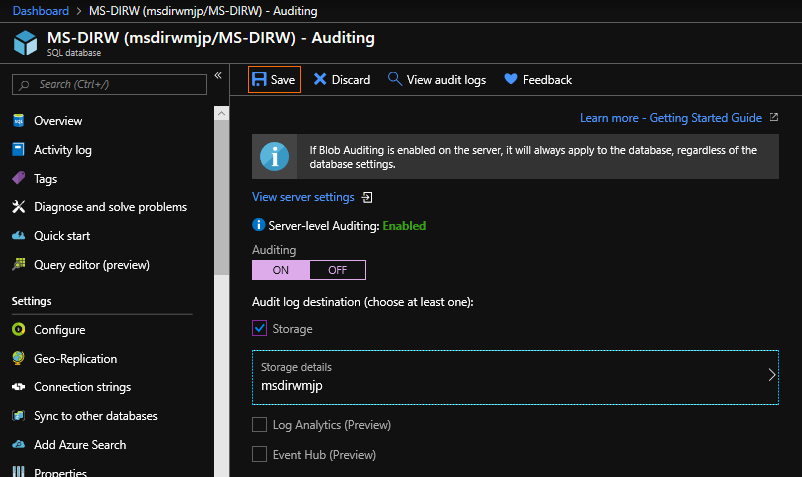


*Task 3: Configure database Auditing*

1. From the SQL database Overview page.
2. At the bottom of the database Overview page select “Database features”, then select “Auditing”
3. Fill in the following fields:
   1. Auditing: ON
   2. Audit log destination: Storage
   3. Storage Details as follows:
      1. Subscription: Select the subscription you are using for the workshop
      2. Storage account: Select the “msdirw” storage account you created (with the unique name)
      3. Retention: 7 days



* + 1. Click OK
  1. Click Save

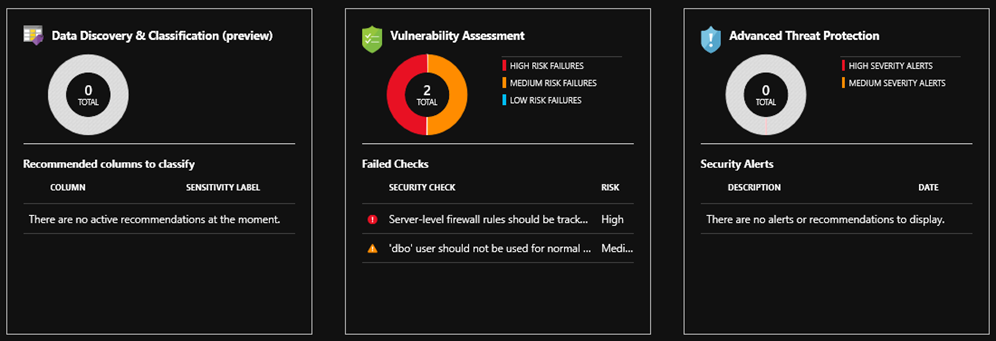


The remaining database features we will not be configuring are:

**Geo-Replication** - we will not need to configure this for our workshop however it is strongly recommended that this option be configured in production deployments to safeguard your database from events such as a regional disaster.

**Dynamic Data Masking** - this allows us to set rules to mask data from non-admin users, such as masking identification, credit cards, tax numbers, email addresses, etc.

**Advanced Data Security** - Advanced Treat Detection, Vulnerability Assessment and Data Classification

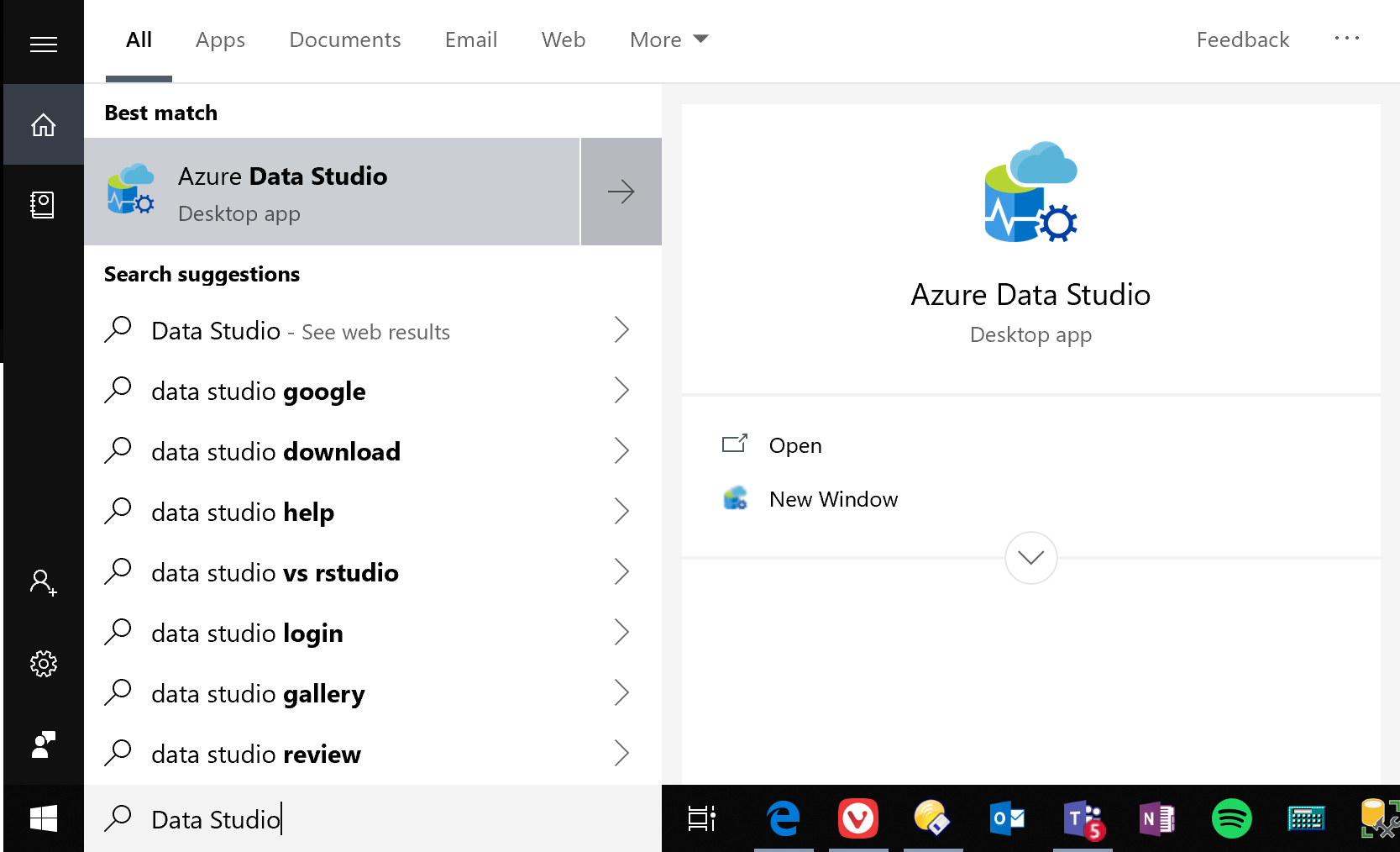


Exercise 3: Create the Azure SQL Database Artifacts to Hold WDI Data

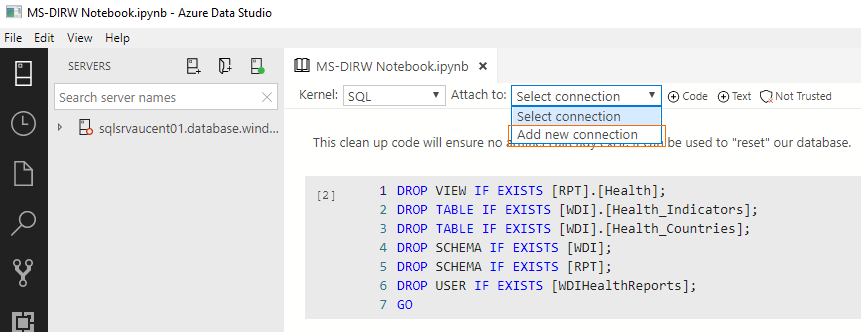
We will use a TSQL notebook to execute on our new database to create the tables and views we require for our analytics and reporting data store.

*Task 1: Execute a TSQL script against our database using* *Azure Data Studio*

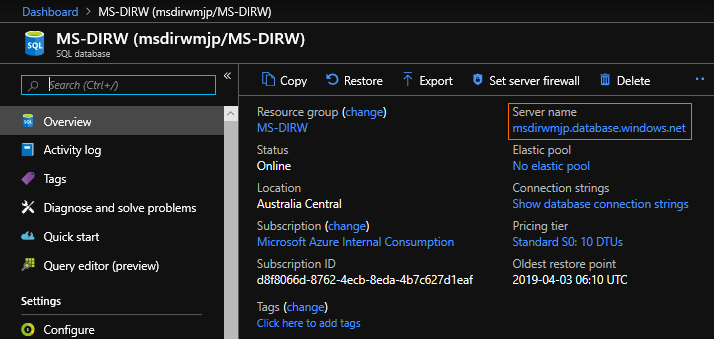
1. Start Azure Data Studio



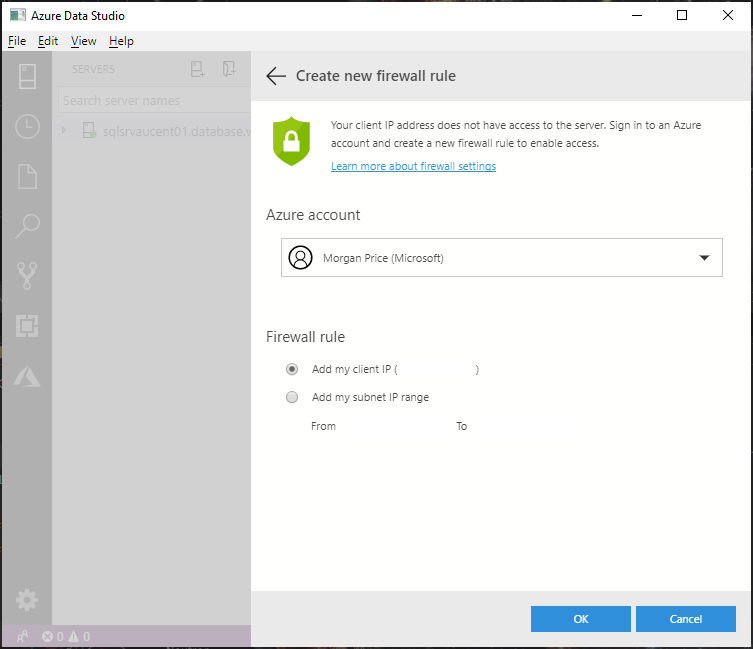
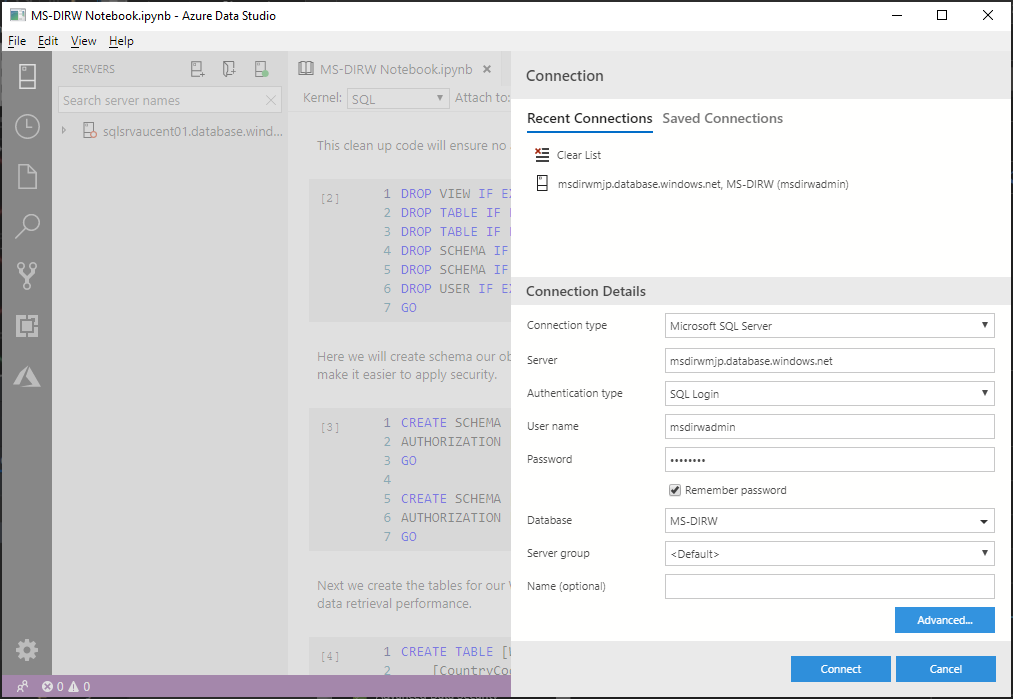
1. Use the File menu to open the SQL notebook “C:\MS-DIRW\MS-DIRW Notebook.ipynb”
2. Click on the “Select Connection” dropdown and select the “Add New Connection” option



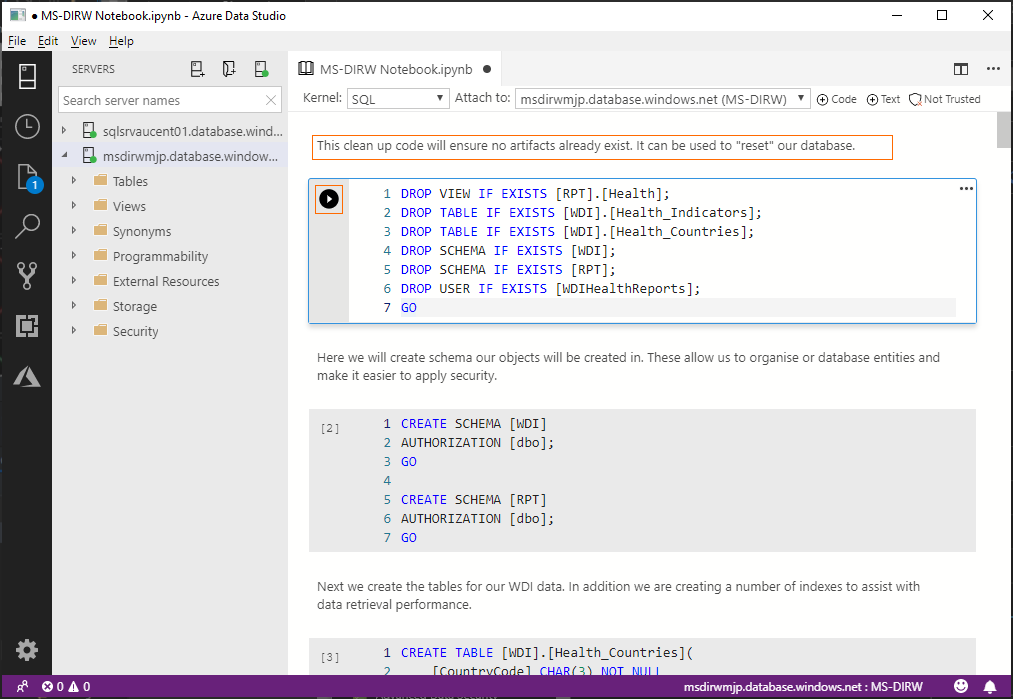
1. Use the following connection details (we are using the Server Administrator credentials we created when we configured the server during the database deployment task):
   1. Connection type: Microsoft SQL Server
   2. Server: Copy your “Server name” from the SQL database Overview page



* 1. Authentication type: SQL Login
  2. User name: msdirwadmin
  3. Password: pa$$w0rd
  4. Remember password: tick
  5. Database: MS-DIRW, when you select this dropdown Azure Data Studio will connect to the database. You will be required to authenticate your account and will then be prompted to add your IP address to the server firewall.



1. Click “Connect”
2. Work through each cell in the notebook, read the markdown text and execute each cell. Check the output to ensure each cell runs successfully



## Lab 04: Deploy Azure Data Factory Solution

In this lab we will deploy and configure an instance of Azure Data Factory. We will create connections to our data sources (Azure Blob Storage, Azure SQL Database), we will then define our datasets (WDI CSV files, WDI SQL tables). From there we can create a pipeline to ETL our CSV file data into our SQL tables to create or analytics and reporting data store.

Prerequisites

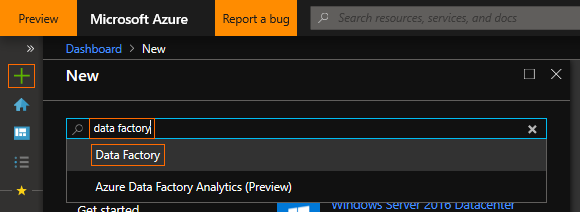
* Previous labs completed successfully

Exercise 1: Deploy an Azure Data Factory Instance

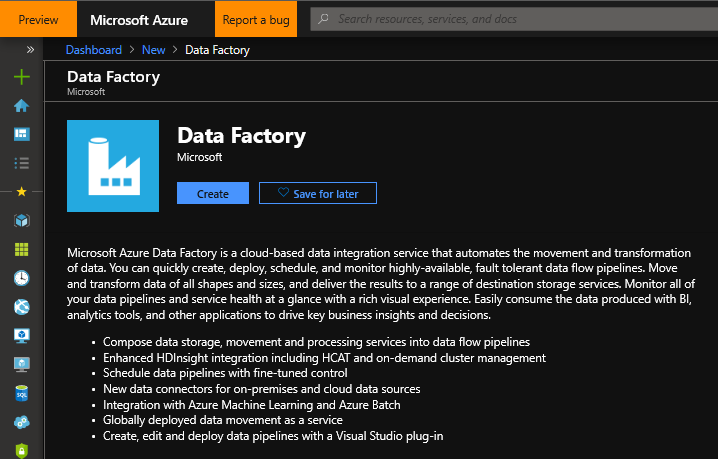
We will deploy an Azure Data Factory Instance to develop our ETL capability.

*Task 1: Deploy Azure Data Factory*

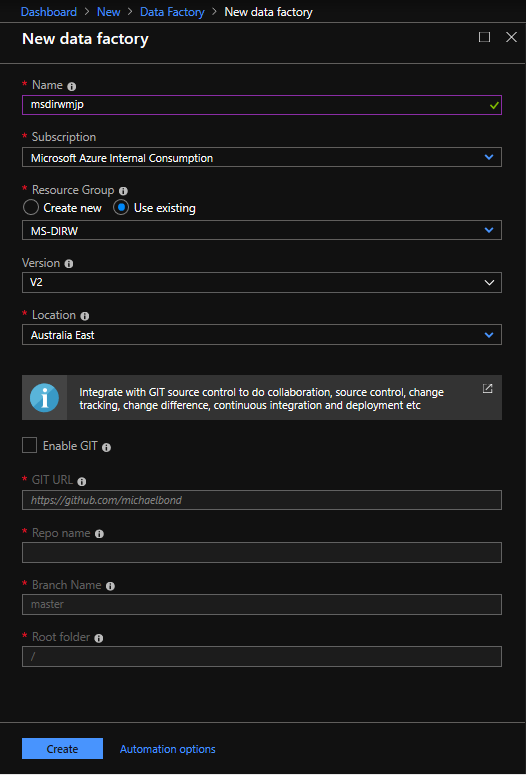
1. Start from your MS-DIRW dashboard.
2. Click on “Create a resource”, type “data factory” in the search box and select the “Data Factory” entry.



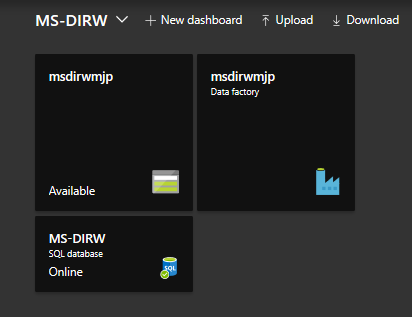
1. Click “Create”



1. Fill in the following fields:
   1. Name: Enter “msdirw” and append your initials, e.g. “msdirwmjp”. This name must be globally unique so add additional information if your chosen name has already been used
   2. Subscription: Select your subscription
   3. Resource group: Select the Resource Group you created when you created your Storage Account in Lab 02. The suggested name was “msdirw” appended with your initials, e.g. “msdirwmjp”. Remember this name had to be globally unique.
   4. Version: V2
   5. Location: “Australia East”
   6. Enable GIT: Unchecked



1. Click “Create” to deploy your new Azure Data Factory
2. When complete click on the notification (access using the little bell icon at the top right) and then click the “Pin to dashboard” button, this will make it easier to access this resource later
3. Click on the “Dashboard” link in the top left navigation path to get back to the dashboard
4. You should now see your new Azure Data Factory on your Dashboard

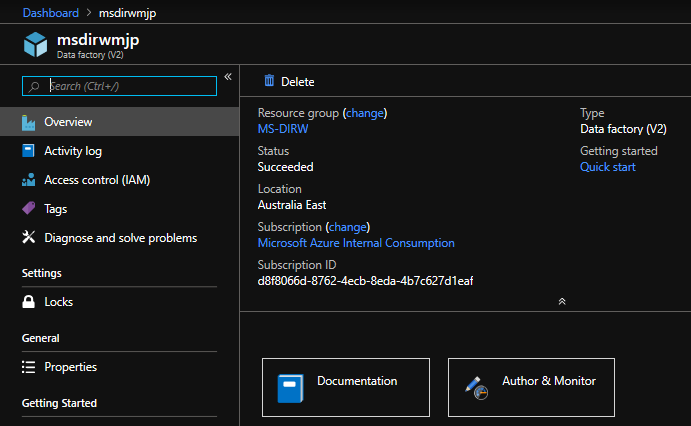


Exercise 2: Create Azure Data Factory Connections

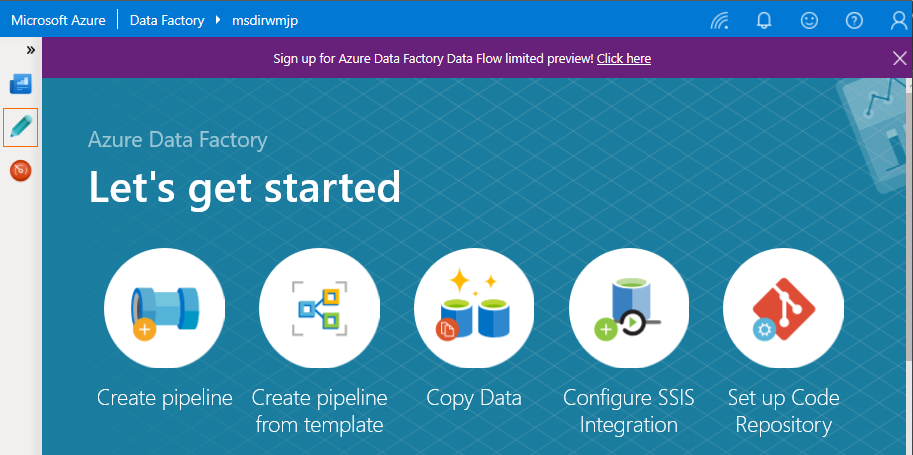
These connections will establish connectivity to our data sources.

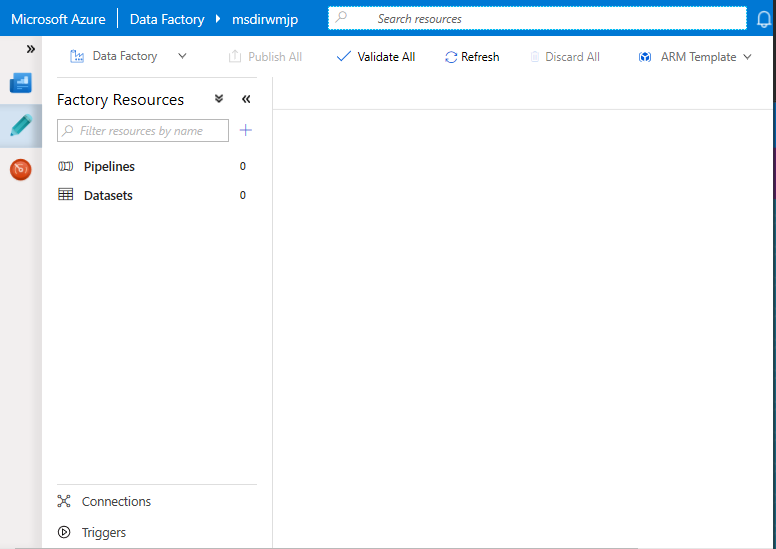
*Task 1: Open you Azure Data Factory workbench*

1. Start from your MS-DIRW dashboard and click on the Data Factory resource
2. Select “Author & Monitor”



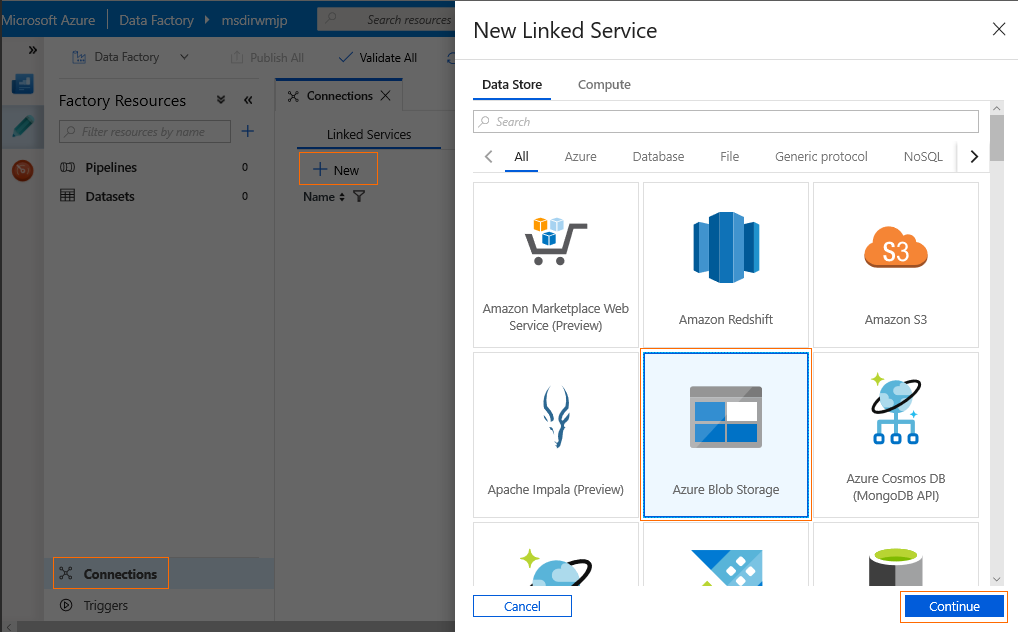
1. We are presented with a “Let’s get started” launch pad. From here we want to launch our workspace by clicking on the “Author” icon, the pencil towards the top left



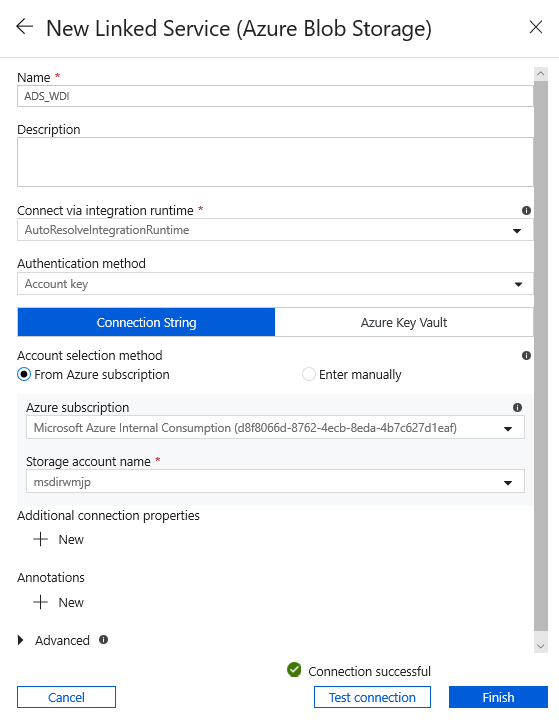


*Task 2: Create a connection to our Azure Blob Storage*

1. From your ADF workbench
2. Click on the “Connections” icon towards the lower left of the ADF workspace, under the “Linked Services” tab click “+ New”.
3. From the “Data Store” flyout select “Azure Blob Storage” and then select “Continue”



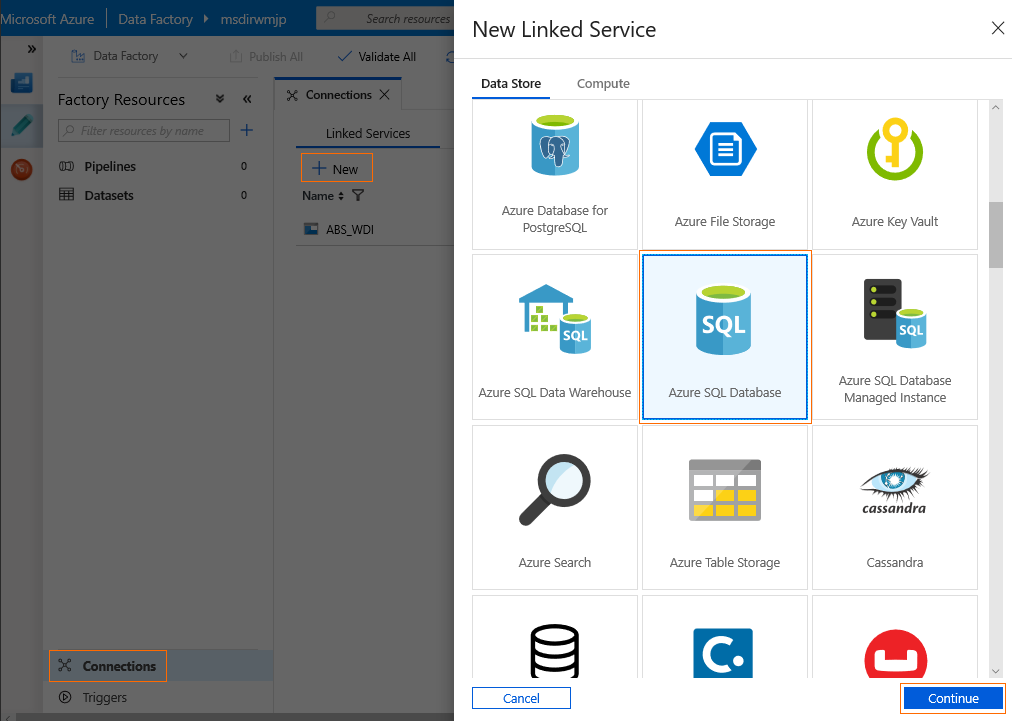
1. Fill the “New Linked Service (Azure Blob Storage)” connection details below, if a field is not listed use the default value – refer to the following screen shot:
   1. Name: “ABS\_WDI”
   2. Azure Subscription: Select your subscription
   3. Storage account name: Select the Storage Account you created in Lab 02. The suggested name was “msdirw” appended with your initials, e.g. “msdirwmjp”. Remember this name had to be globally unique.



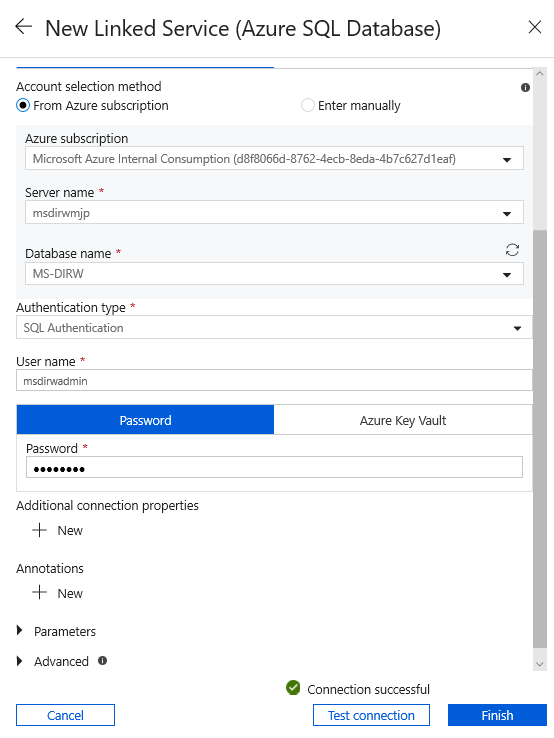
1. ADF will obtain and store the required access keys for you. Click the “Test connection” button to validate the connection
2. Click “Finish”

*Task 3: Create a connection to our Azure SQL Database*

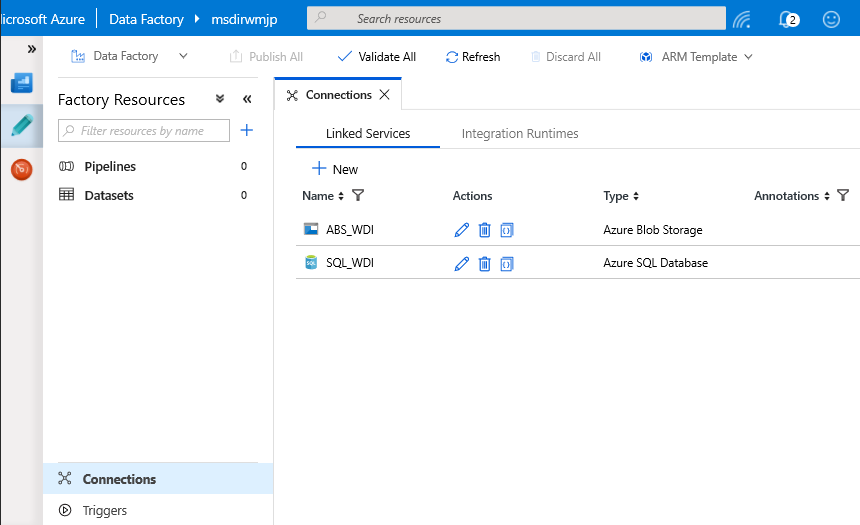
1. From your ADF workbench
2. Click on the “Connections” icon towards the lower left of the ADF workspace, under the “Linked Services” tab click “+ New”.
3. From the “Data Store” flyout select “Azure SQL Database” and then select “Continue”



1. Fill the “New Linked Service (Azure SQL Database)” connection details below, if a field is not listed use the default value – refer to the following screen shot:
   1. Name: “SQL\_WDI”
   2. Azure Subscription: Select your subscription
   3. Server name: Select the Server you created in Lab 03. The suggested name was “msdirw” appended with your initials, e.g. “msdirwmjp”. Remember this name had to be globally unique
   4. Database name: Select “MS-DIRW”
   5. User name: “msdirwadmin”
   6. Password: “pa$$w0rd”



1. ADF will obtain and store the required access keys for you. Click the “Test connection” button to validate the connection
2. Click “Finish”

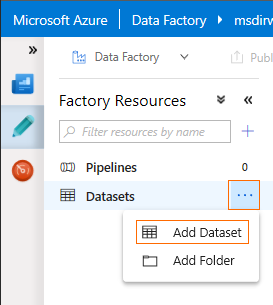


Exercise 3: Create Azure Data Factory Datasets

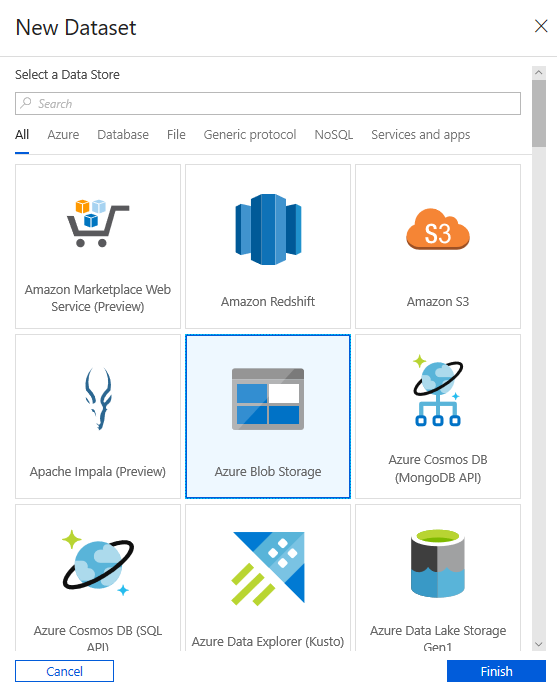
These datasets will define the structure of our data.

*Task 1: Create a dataset representing the WDI countries CSV data stored in our Azure Storage Blob*

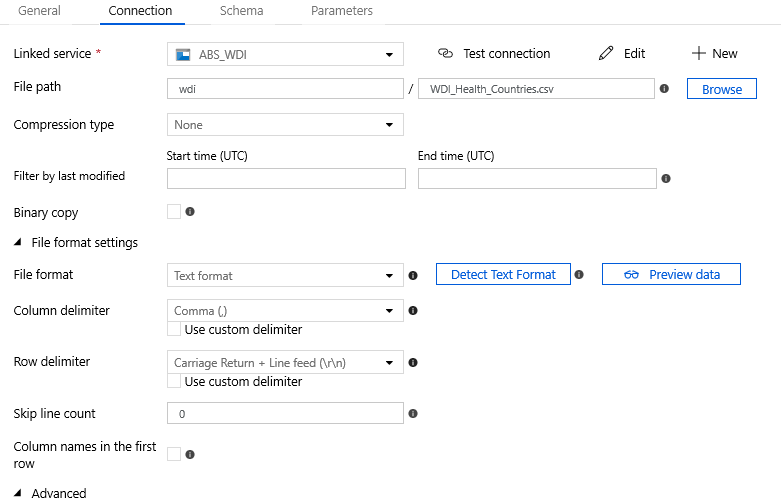
1. From your ADF workbench
2. Click on the “…” menu beside the Datasets entry and select “Add Dataset” from the dropdown



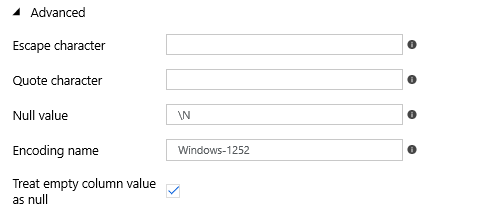
1. Select “Azure Blob Storage” and click on “Finish”



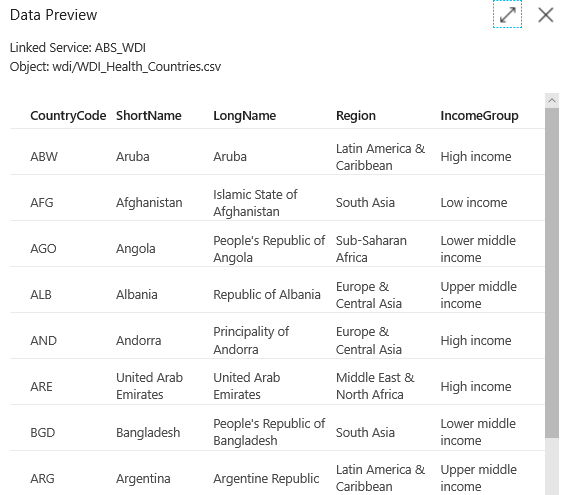
1. Under the “General” tab, Name: “ABS\_WDI\_Health\_Countries”
2. Under the “Connection” tab fill in the following fields, if a field is not listed use the default value – refer to the following screen shot:
   1. Linked service: Select “ABS\_WDI”
   2. File path: User the “Browse” button and select the “WDI\_Health\_Countries.csv” file in the “wdi” container
   3. Column names in the first row: Checked



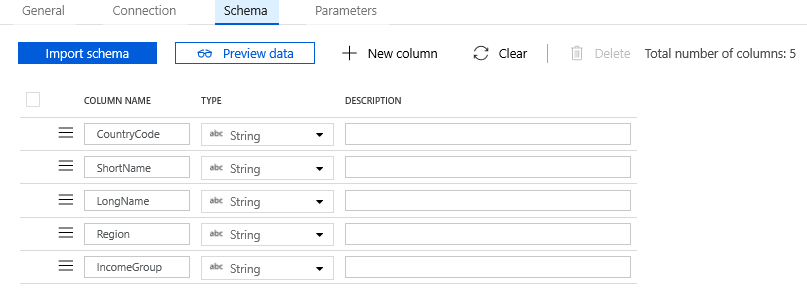
* 1. Advanced: Click on “Advanced” to expand these options:
     1. Encoding name: “Windows-1252”, this will ensure accented characters in our CSVs are handled correctly



1. Click on the “Preview data” button and confirm the preview looks like the following screen shot



1. Under the “Schema” tab, click on the “Import schema” button

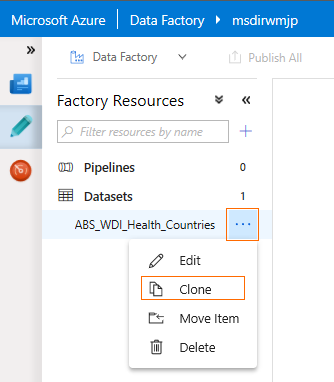


1. Click the “Publish All” button at the top of the workbench to commit changes to the Data Factory

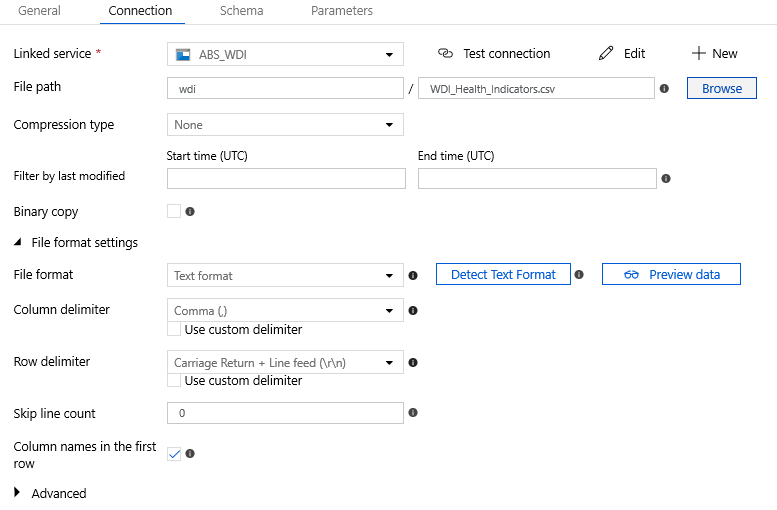


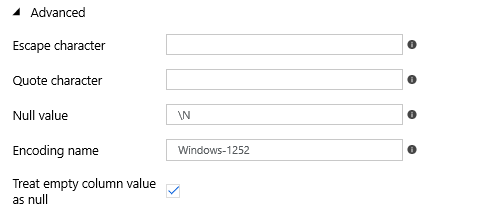
*Task 2: Create a dataset representing the WDI indicators CSV data stored in our Azure Storage Blob by cloning the WDI countries CSV dataset*

1. From your ADF workbench
2. Click on the “…” menu beside the “ABS\_WDI\_Health\_Countries” Dataset entry and select “Clone” from the dropdown

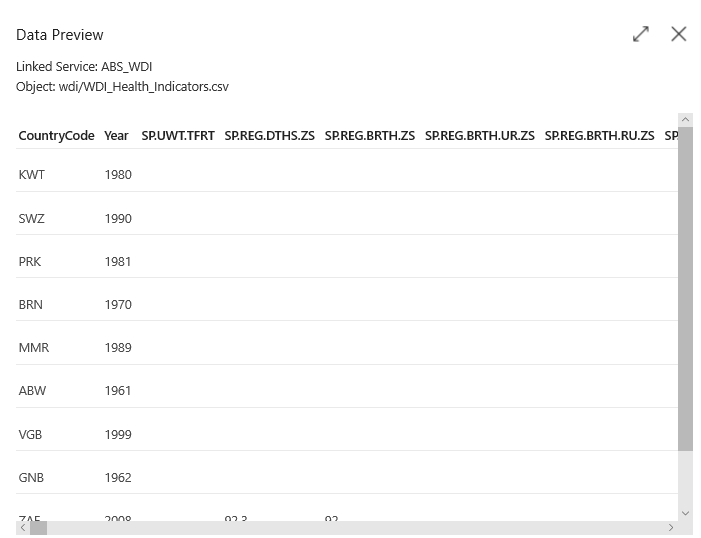


1. Under the “General” tab, Name: “ABS\_WDI\_Health\_Indicators”
2. Under the “Connection” tab fill in the following fields, if a field is not listed use the default value – refer to the following screen shot:
   1. File path: User the “Browse” button and select the “WDI\_Health\_Countries.csv” file in the “wdi” container

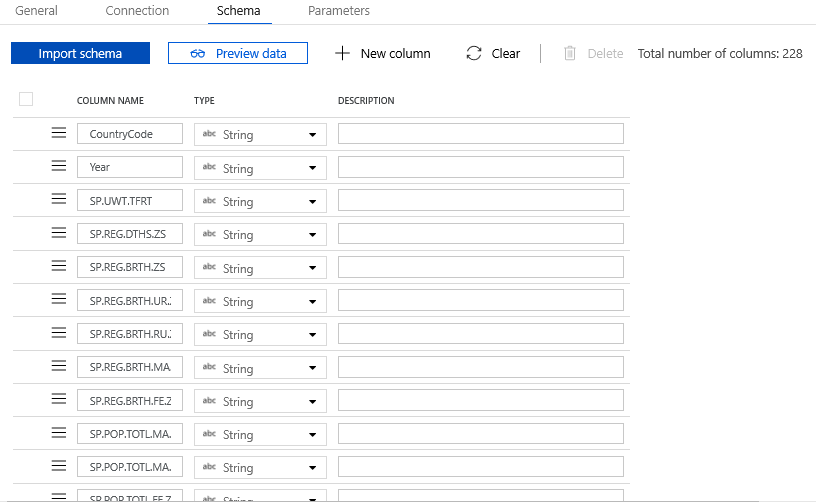




1. Click on the “Preview data” button and confirm the preview looks like the following screen shot



1. Under the “Schema” tab, first click on the “Clear” button to remove the existing entries and then click on the “Import schema” button

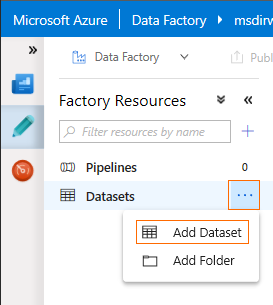


1. Click the “Publish All” button at the top of the workbench to commit changes to the Data Factory



*Task 3: Create a dataset representing the WDI countries table in our Azure SQL Database*

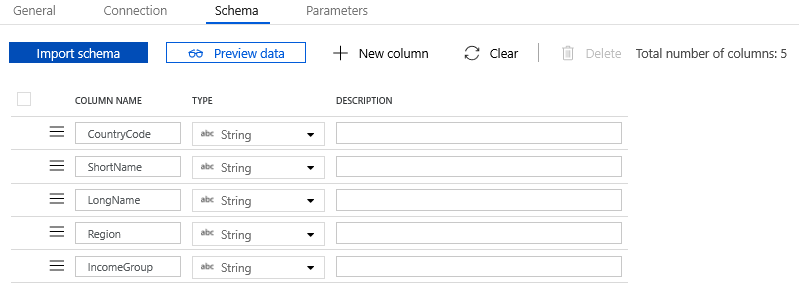
1. From your ADF workbench
2. Click on the “…” menu beside the Datasets entry and select “Add Dataset” from the dropdown



1. Select “Azure Blob Storage” and click on “Finish”



1. Under the “General” tab, Name: “SQL\_WDI\_Health\_Countries”
2. Under the “Connection” tab fill in the following fields:
   1. Linked service: Select “SQL\_WDI”
   2. Table: Select “[WDI].[Health\_Countries]”
3. Under the “Schema” tab, click on the “Import schema” button

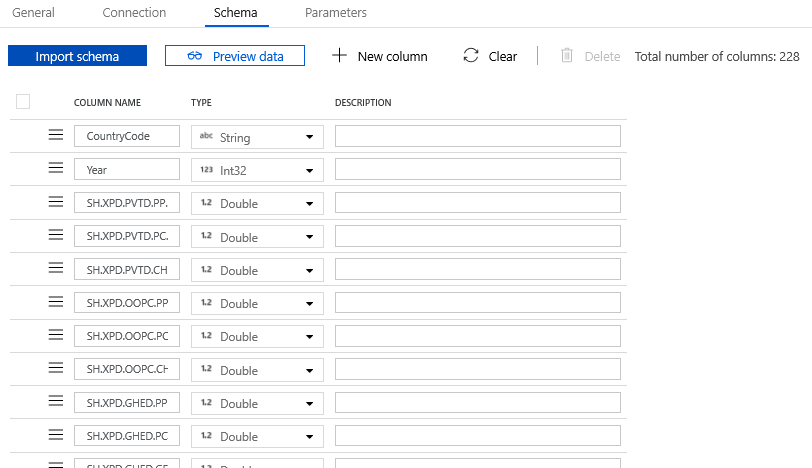


1. Click the “Publish All” button at the top of the workbench to commit changes to the Data Factory



*Task 4: Create a dataset representing the WDI indicators table our Azure SQL Database by cloning the WDI indicators SQL dataset*

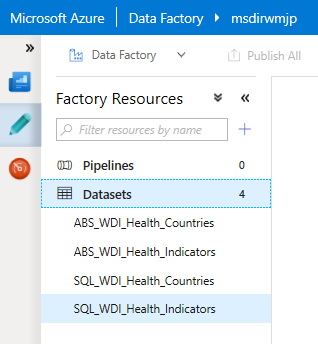
1. From your ADF workbench
2. Click on the “…” menu beside the “SQL\_WDI\_Health\_Countries” Dataset entry and select “Clone” from the dropdown
3. Under the “General” tab, Name: “SQL\_WDI\_Health\_Indicators”
4. Under the “Connection” tab fill in the following fields:
   1. Linked service: Select “SQL\_WDI”
   2. Table: Select “[WDI].[Health\_Indicators]”
5. Under the “Schema” tab, first click on the “Clear” button to remove the existing entries and then click on the “Import schema” button



1. Click the “Publish All” button at the top of the workbench to commit changes to the Data Factory



1. Confirm your ADF resource list looks like the following and all changes are committed

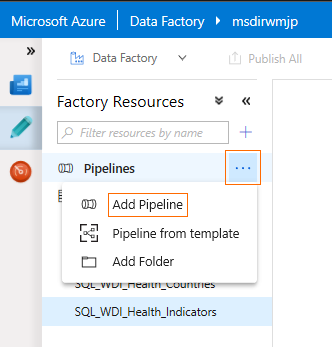


Exercise 4: Create and run the Azure Data Factory Pipeline

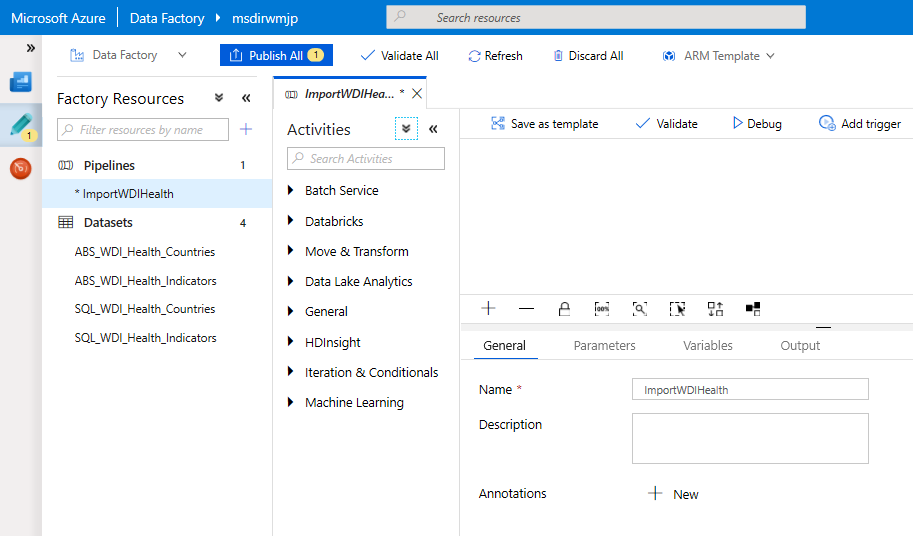
This pipeline will define the data movement phase of our ADF solution. We will be moving our CSV data stored in Azure Blob Storage to our Azure SQL tables making it ready for analysis and reporting.

*Task 1: Create an ADF pipeline*

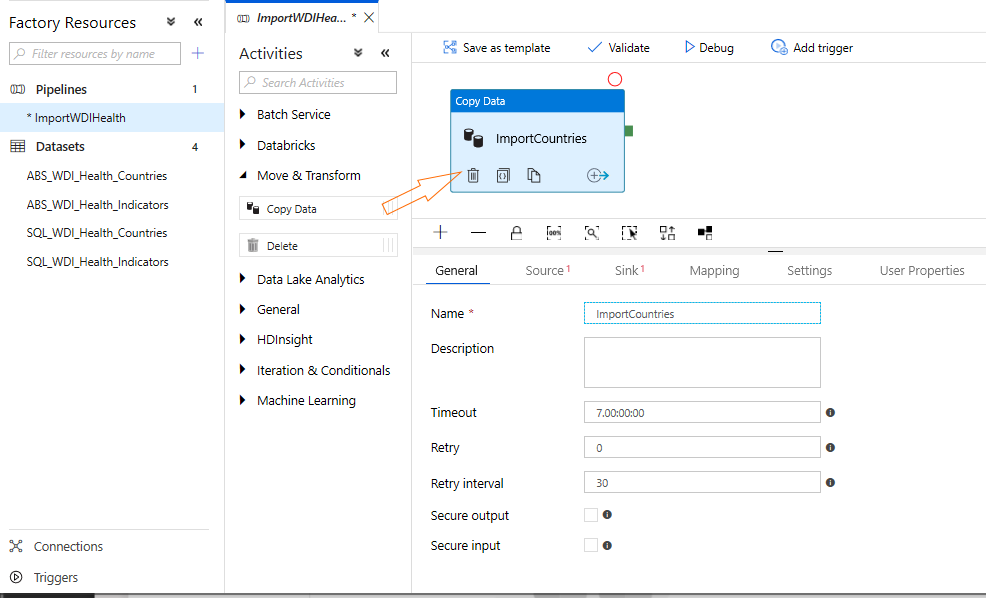
1. From your ADF workbench
2. Click on the “…” menu beside the Pipelines entry and select “Add Pipeline” from the dropdown



1. Under the “General” tab, Name: “ImportWDIHealth”



1. Expand “Move & Transform” entry under the “Activities” list and drag an instance of the “Copy Data” activity on to workspace
2. The properties pane should now reflect the properties of the Copy activity, if not click on the Copy activity you created in the workspace
3. Under the General tab, Name: “ImportCountries”



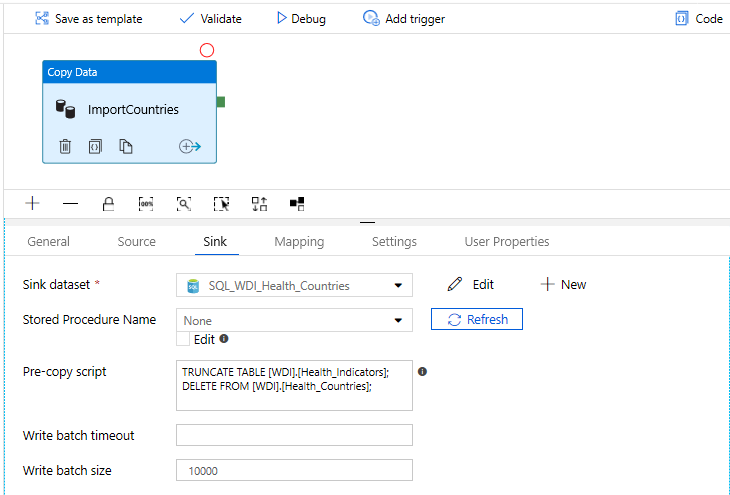
1. Under the “Source” tab, Source dataset: Select “ABS\_WDI\_Health\_Countries”



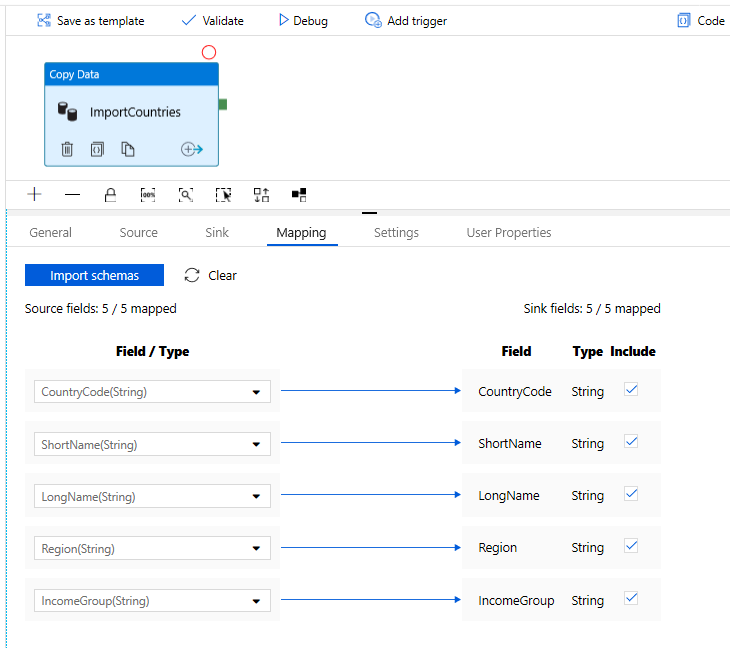
1. Under the “Sink” tab:
   1. Sink dataset: “SQL\_WDI\_Health\_Countries”
   2. Stored Procedure Name: Select “None”
   3. Pre-copy script: Copy and paste the 2 lines below, this script ensures the SQL tables have no data in them so this pipeline can easily be re-run if necessary

TRUNCATE TABLE [WDI].[Health\_Indicators];

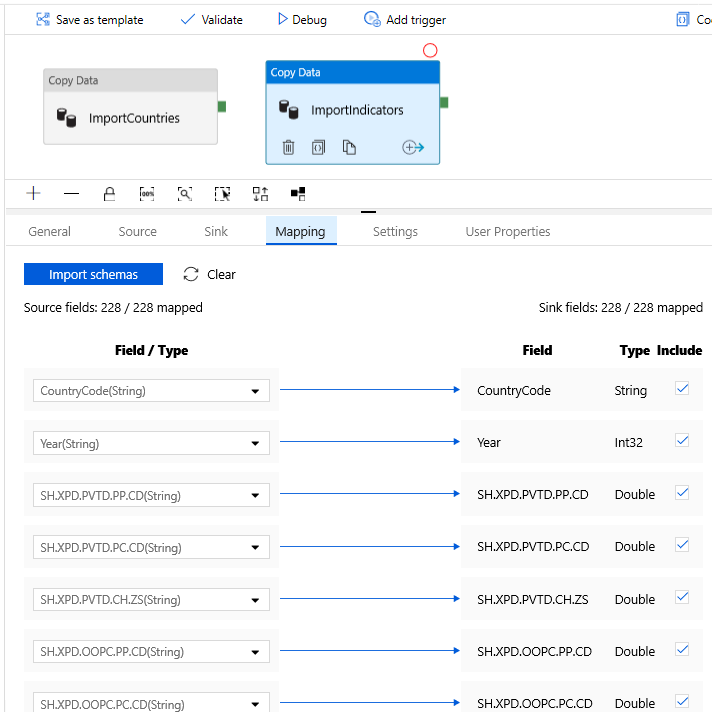
DELETE FROM [WDI].[Health\_Countries];



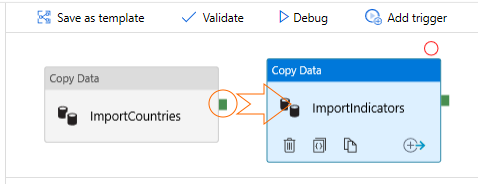
1. Under the “Mapping” tab click the “Import schemas” button



1. Drag another instance of the “Copy Data” activity and position it to the right of the “ImportCountries” activity we just configured
2. Under the General tab, Name: “ImportIndicators”
3. Under the “Source” tab, Source dataset: Select “ABS\_WDI\_Health\_Indicators”
4. Under the “Sink” tab:
   1. Sink dataset: “SQL\_WDI\_Health\_Indicators”
   2. Stored Procedure Name: Select “None”
5. Under the “Mapping” tab click the “Import schemas” button



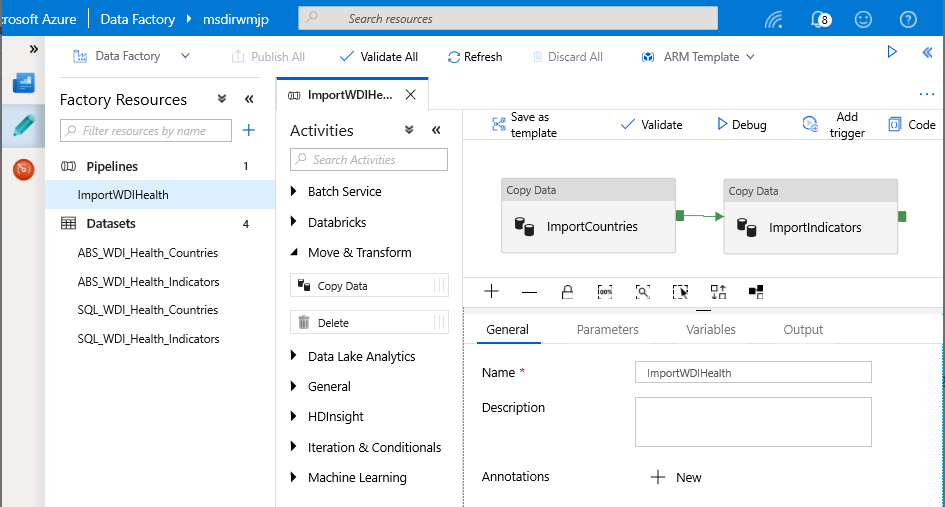
1. Create a precedence flow by dragging the from the green handle on the “ImportCountries” activity and releasing over the “ImportIndicators” activity



1. Click the “Publish All” button at the top of the workbench to commit changes to the Data Factory

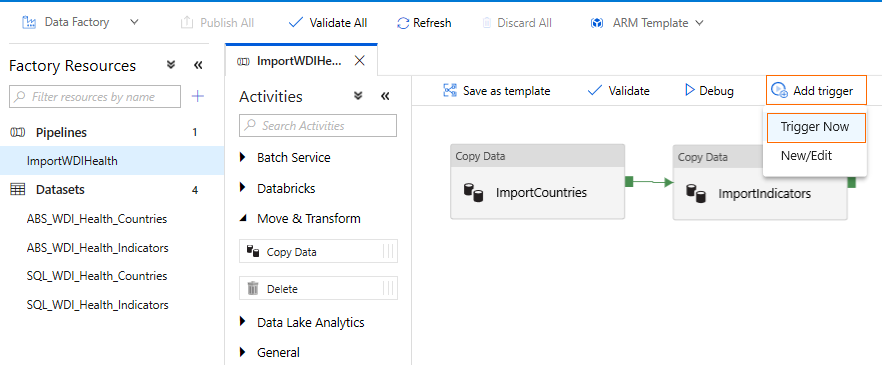


1. Ensure the ADF workspace is as below

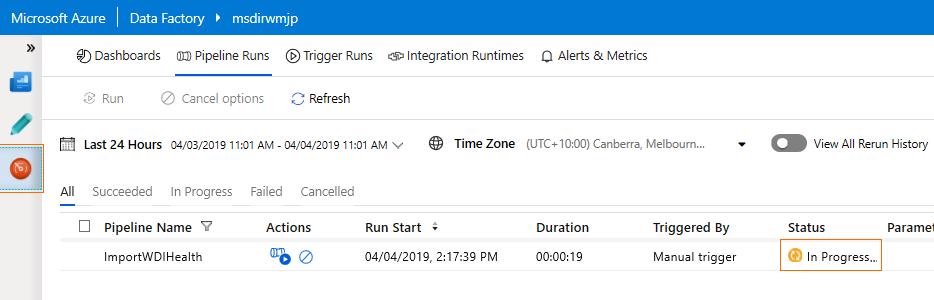


*Task 2: Run the ADF pipeline*

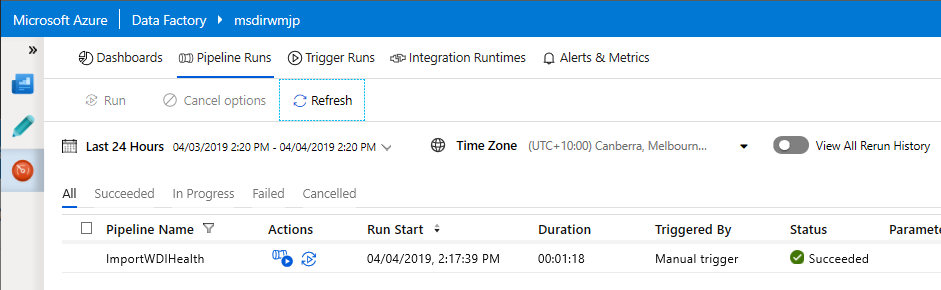
1. From your ADF workbench
2. Open the “ImportWDIHealth” pipeline in the workspace, click on the “Add trigger” button and select “Trigger Now” from the dropdown



1. Select the “Monitor” icon located toward the upper left of the workbench, the Status should be “In Progress”



1. Click on the “Refresh” button until the run has completed – this should take a little over a minute

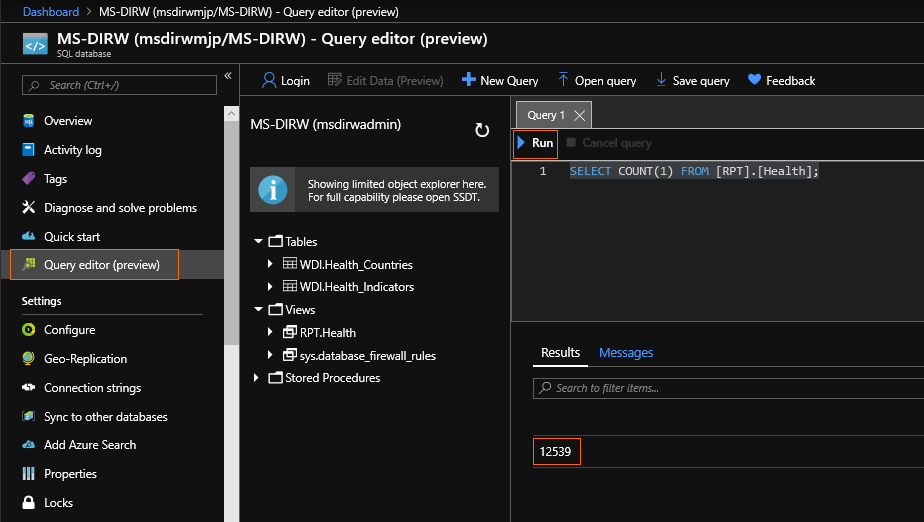


*Task 3: Validate the data has loaded into the Azure SQL database*

1. From the MS-DIRW dashboard select the MS-DIRW SQL database
2. Select “Query editor (preview)” from the menu list on the left
3. Enter the following credentials, Login: “msdirwadmin”, Password: “pa$$w0rd”, click “OK”
4. Paste the following query into the query editor:

SELECT COUNT(1) FROM [RPT].[Health];

1. Click “Run”
2. Confirm the result is 12539



## Lab 05: Provision Azure Databricks for Data Analysis and Prediction

TODO: Lab overview

In this lab we will provision an instance of an Azure Databricks. We will then import the notebook to Azure Databricks workspace. This notebook contains the R code which ingests data from Azure SQL DB and builds a prediction model based on Decision trees. The predictions are written back to Azure SQL DB.

Prerequisites

* Previous labs completed successfully

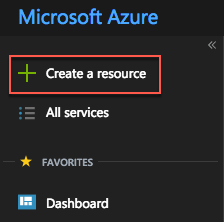
Exercise 1: Setup Azure Databricks Workspace

Duration: 20 minutes

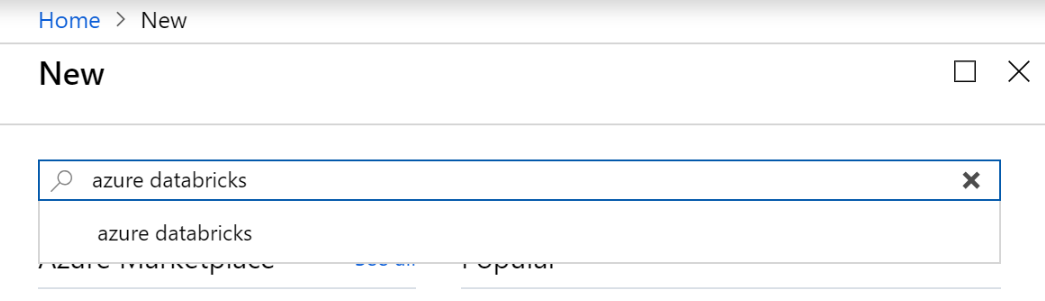
In this exercise, you will setup your Azure Databricks account and Workspace

#### Task 1: Provision Azure Databricks service

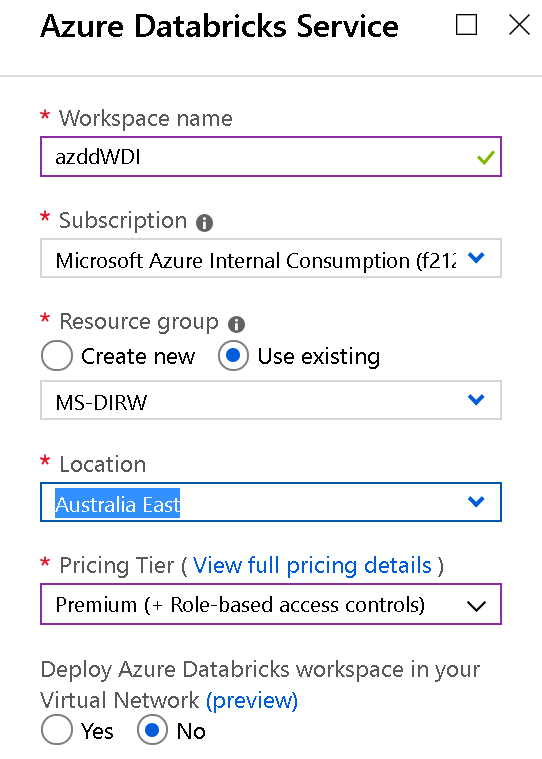
1. Navigate to the Azure Portal.
2. Select **Create a resource**.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/azure-portal-create-a-resource.png)

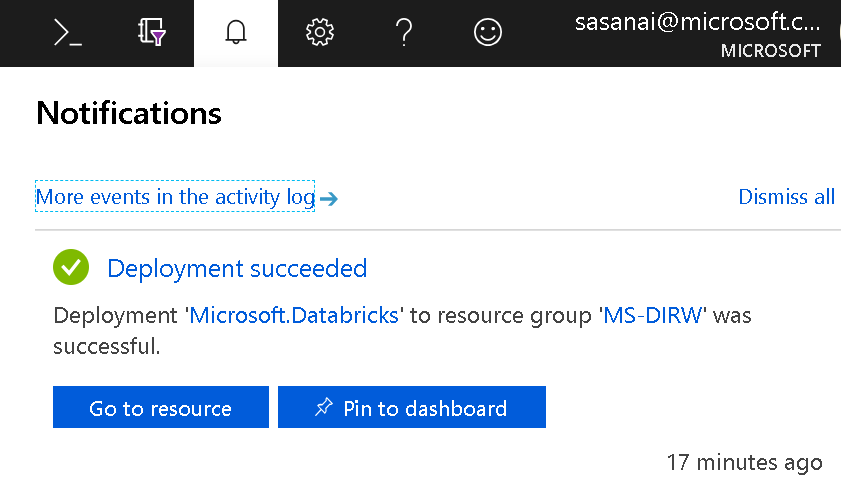
1. In the search box, enter **Azure Databricks** and then select the matching entry that appears with the same name.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image3.png)

1. On the **Azure Databricks** blade, select **Create**.
2. On the **Azure Databricks Service**, enter the following and then select **Create**:
   1. **Workspace Name**: Enter **azddWDI**
   2. **Subscription**: Choose your Azure subscription.
   3. **Resource group**: Choose “Use existing” and then select **MS-DIRW.**
   4. **Location**: Choose a location near you e.g. **Australia East**
   5. **Pricing Tier**: Select **Premium.**
   6. **Virtual Network**: Select “No” for Deploy Azure Databricks in your Virtual Network



1. When the deployment completes, navigate to your deployed Azure Databricks service “Go to Resource”

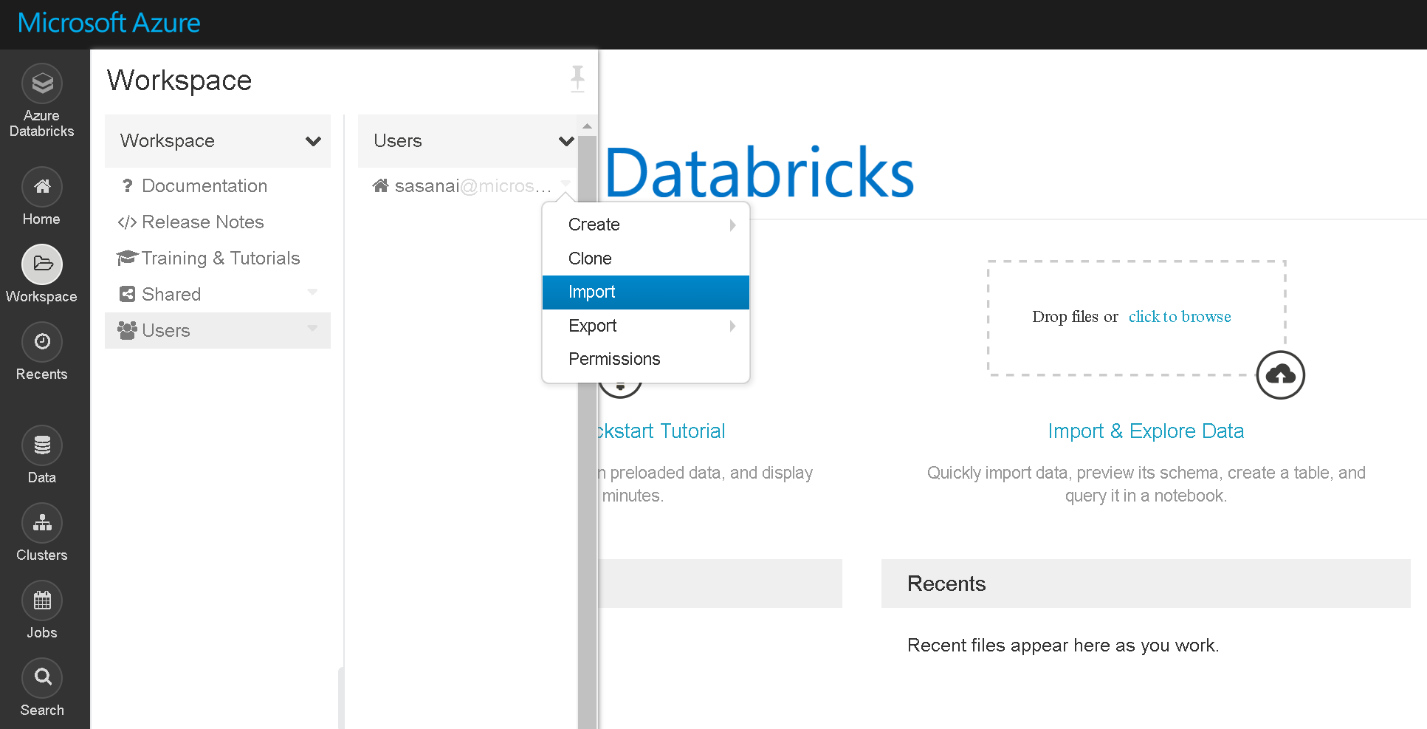


1. Select **Launch Workspace**. If prompted, sign in using the same credentials you used to access the Azure Portal.

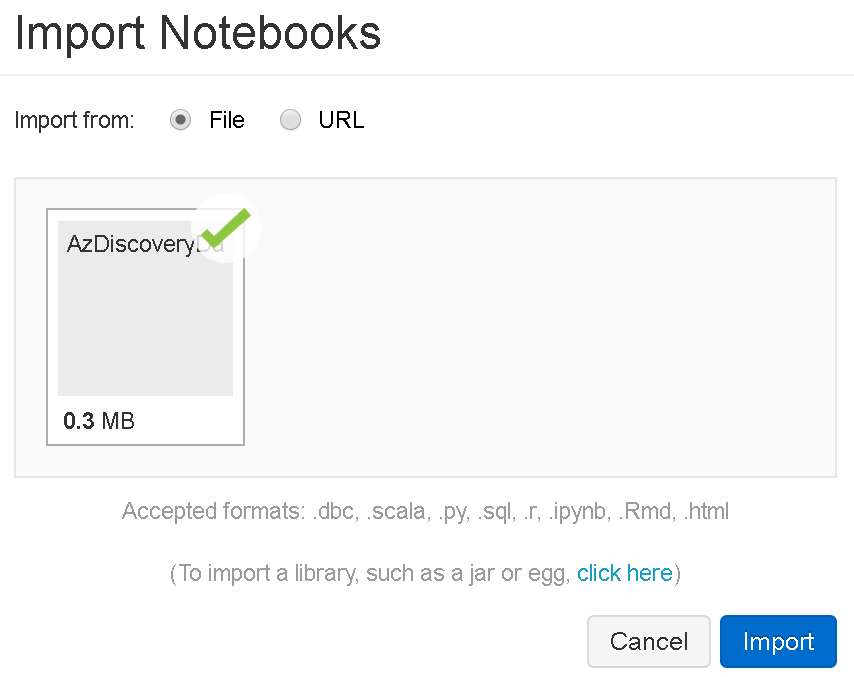
[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image3-2.png)

#### Task 2: Upload the Databricks notebook archive

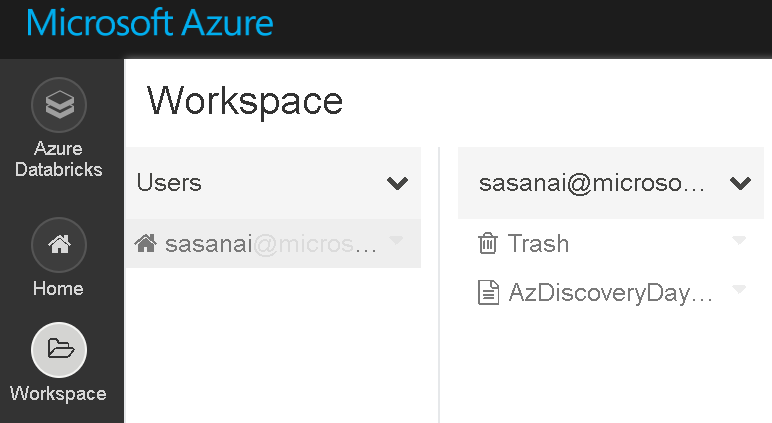
1. Within the Workspace, using the command bar on the left, select Workspace, Users and select your username (the entry with house icon).
2. In the blade that appears, select the downwards pointing chevron next to your name, and select Import.



1. On the Import Notebooks dialog,
   1. Either select the file [AzDiscoveryDayAGD.dbc](https://github.com/MSFT-MJP/MS_DIRW/blob/master/Databricks/AzDiscoveryDayAGD.dbc) from the location “C:\MS-DIRW” where you downloaded it in Exercise 3 of Lab 01.

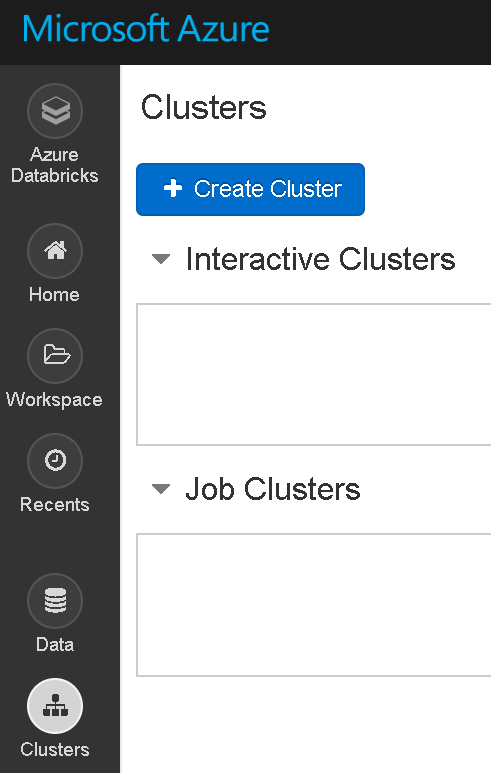


1. Select Import.
2. A Notebook should appear “AzDiscoveryDayAGD”
3. You will use this notebook to complete this lab.

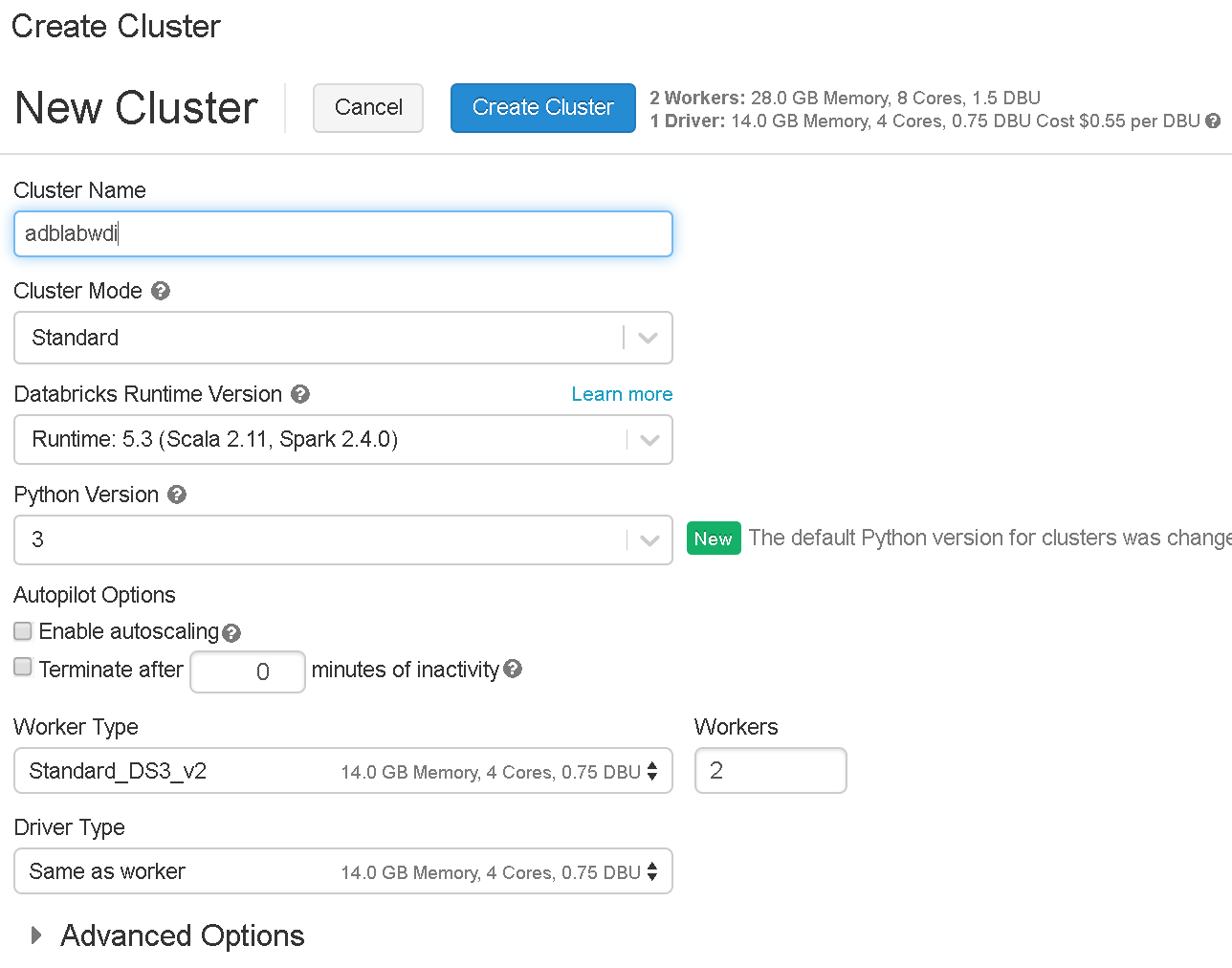


#### Task 3: Provision a cluster

1. Within the Workspace, from the menu on the left, select Clusters.



1. Select **+ Create Cluster**.
2. On the New Cluster page, provide the following:
   1. **Cluster Name**: adblabwdi
   2. **Cluster Mode**: Standard
   3. **Databricks Runtime Version**: 5.3 (includes Apache Spark 2.4.0, Scala 2.11)
   4. **Python Version**: 3
   5. **Worker Type**: Standard\_DS3\_v2
   6. **Driver Type**: Same as worker
   7. **Enable autoscaling**: Unchecked
   8. **Terminate After**: Unchecked
   9. **Workers**: 2

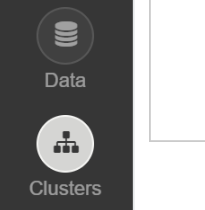


1. Select **Create Cluster**. This will take about 5 minutes to provision your cluster. The cluster will show a State of "Ready" when the cluster is available for use.

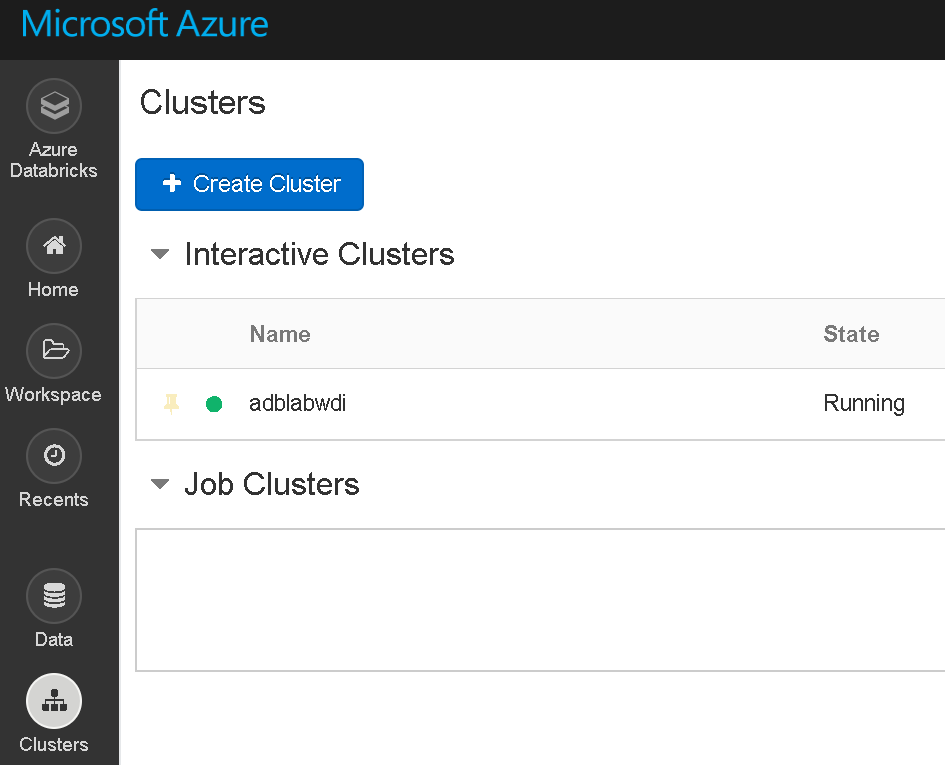
#### Task 4: Install libraries

The notebook you will run depends on certain R libraries like Hmisc and rattle that will need to be installed in your cluster. The following steps walk you through adding these dependencies.

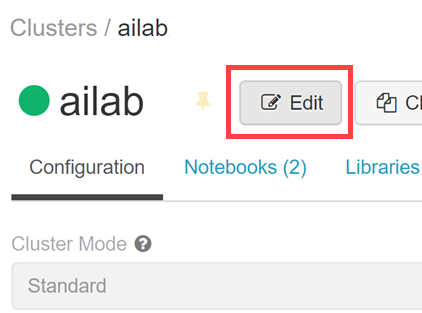
1. From the left-hand menu in your Workspace, select **Clusters**.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image3-4.png)

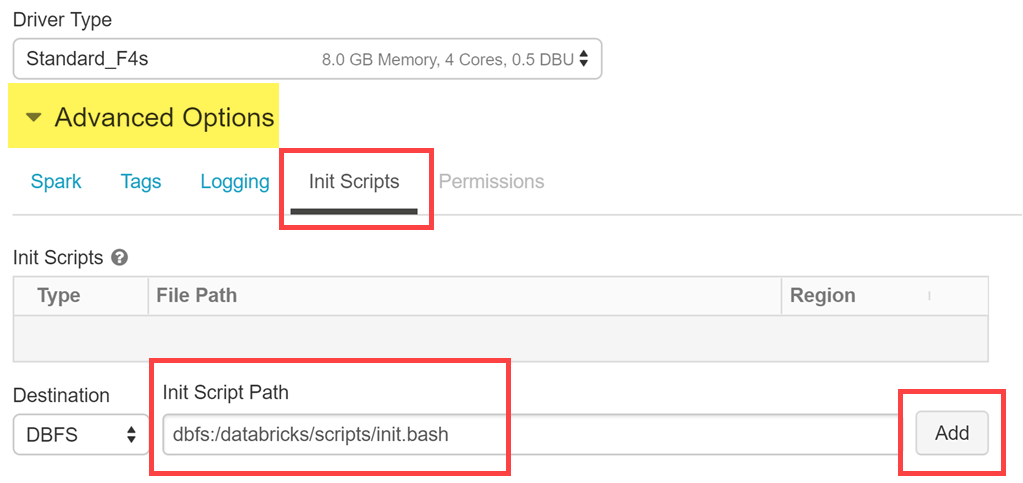
1. In the list of clusters, select your cluster.



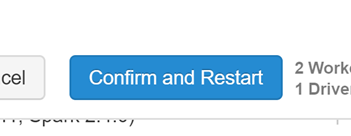
1. Select **Edit** to configure the cluster.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/edit-cluster.png)

1. Expand **Advanced Options** at the bottom, then select **Init Scripts**. Enter the following into the Init Script Path, then select **Add**: dbfs:/databricks/scripts/init.bash

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/init-scripts.png)

1. Select **Confirm and Restart** after adding the Init Script Path.

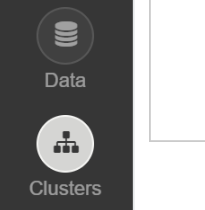
[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/confirm-and-restart.png)

### Task 2: Read through and execute the Summarization notebook

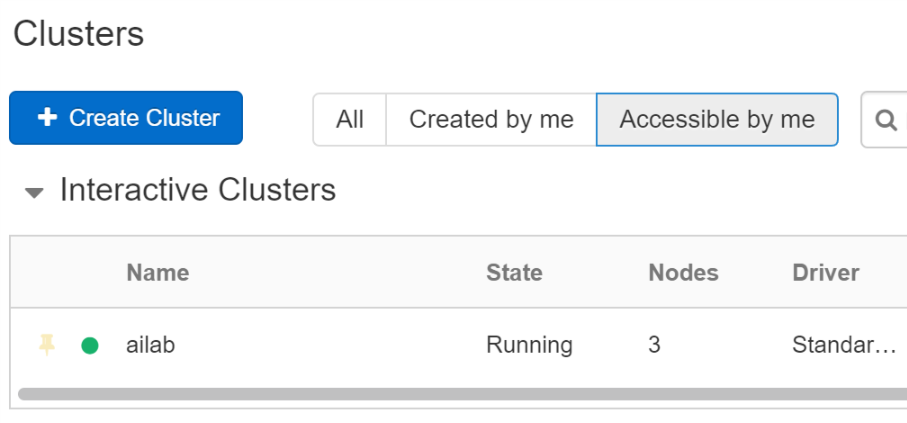
1. Within the Workspace, select the Workspace item in the menu and navigate to the folder where you uploaded the Databricks Archive (which should be [your-name/AI-lab]), and select the notebook called 01 Summarize. This will open the notebook so you can read and execute the code it contains.
2. Read the instructions at the top of the notebook, and execute the cells as instructed. Remember you can use SHIFT + ENTER to execute the currently selected cell, and if you do not have a cluster attached, you will be prompted to attach to the cluster you recently deployed.

### Task 3: Provision the Azure Machine Learning Workspace and Create the Summarization service

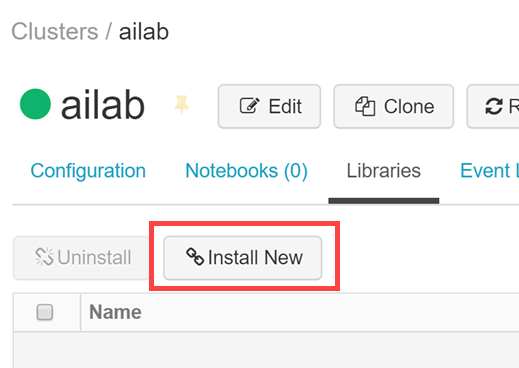
1. From the left-hand menu in your Workspace, select **Clusters**.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image3-4.png)

1. In the list of clusters, select your cluster.

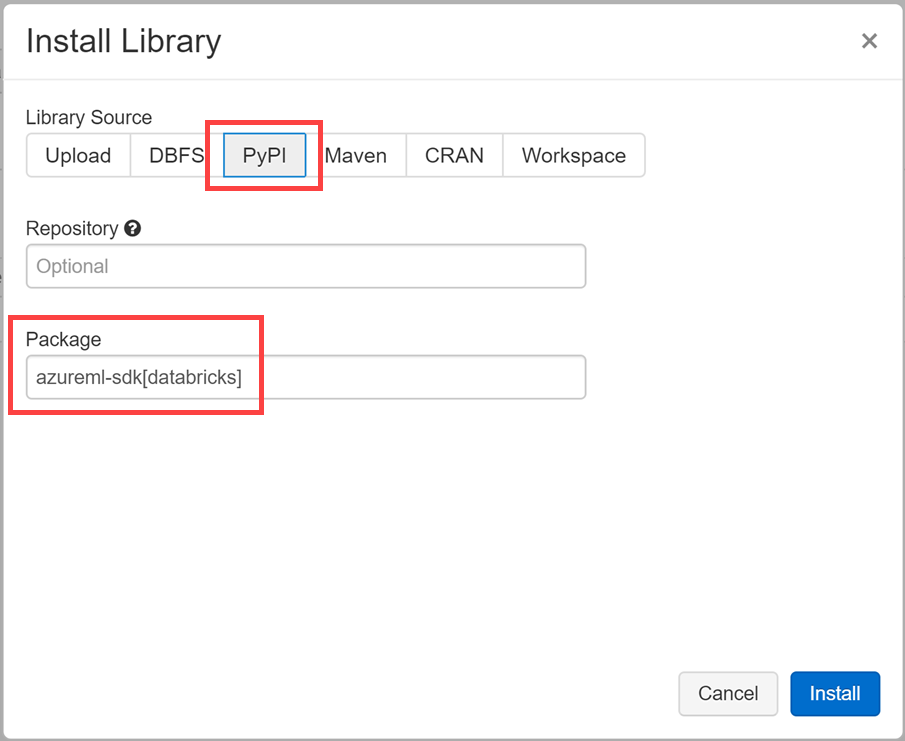
[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image3-5.png)

1. Select the **Libraries** tab and then select **Install New**.

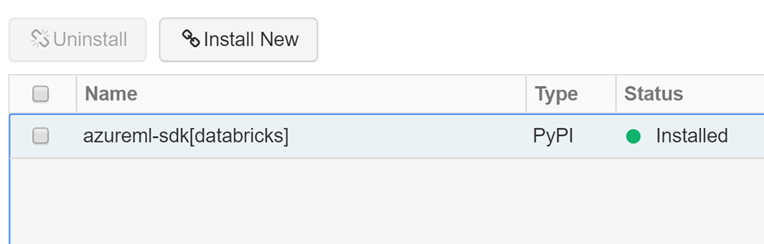
[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image3-6.png)

**Note**: There are interface updates being deployed, if you do not see the Install New button, instead go to the Azure Databricks menu option in your Workspace (the very top option on the left) and select Import Library. Then select a source of **Upload Python Egg or PyPi** and then provide the Package name specified in the following steps in the PyPi Name text box. Then in the Status on running clusters list, check the checkbox Attach that is listed to the left of your cluster's name to install the library on your cluster. When successful the Status should read Attached.

1. In the Library Source, select **PyPi** and in the Package text box type azureml-sdk[databricks] and select Create.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/install-azureml-sdk.png)

1. An entry for azureml-sdk[databricks] will appear in the list with a status of installing followed by installed. All the other required libraries are installed through the cluster init script you added.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/cluster-installed-library.png)

1. Within the Workspace, select the Workspace item in the menu and navigate to the folder where you uploaded the Databricks Archive (which should be [your-name/AI-lab]), and select the notebook called 02 Deploy Summarizer Web Service. This will open the notebook so you can read and execute the code it contains.
2. Read the instructions at the top of the notebook, and execute the cells as instructed. Remember you can use SHIFT + ENTER to execute the currently selected cell, and if you do not have a cluster attached, you will be prompted to attach to the cluster you recently deployed.

Pay attention to the top of the notebook where you are asked to ensure the Azure Machine Learning Python SDK (azureml-sdk[databricks]) is installed. You completed this in steps 3 and 4 above.

## Exercise 3: Create and Deploy a TensorFlow Model

Duration: 60 minutes

In this exercise, you will use TensorFlow to construct and train a simple deep neural network classification model that will classify claim text as belonging to a home insurance claim or an automobile claim. You will then deploy this trained model as a web service.

### Task 1: Create a simple TensorFlow based model

1. Within the Workspace, select the Workspace item in the menu and navigate to the folder where you uploaded the Databricks Archive (which should be [your-name/AI-lab]), and select the notebook called 03 Claim Classification. This will open the notebook so you can read and execute the code it contains.
2. Read the instructions at the top of the notebook, and execute the cells as instructed. Remember you can use SHIFT + ENTER to execute the currently selected cell, and if you do not have a cluster attached, you will be prompted to attach to the cluster you recently deployed.

Pay attention to the top of the notebook where you are asked to ensure the tensorflow and tflearn libraries are installed. If you are running this lab in a hosted environment, they will already be installed. Otherwise, follow the posted instructions to install the libraries, ensuring they are installed and attached to your cluster before you run the cells in the notebook.

### Task 2: Deploy the TensorFlow model

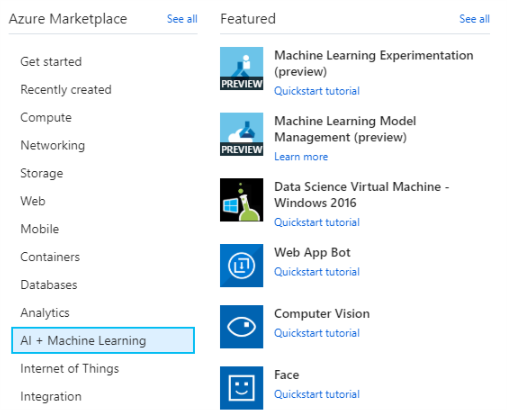
1. Within the Workspace, select the Workspace item in the menu and navigate to the folder where you uploaded the Databricks Archive (which should be [your-name/AI-lab]), and select the notebook called 04 Deploy Classifier Web Service. This will open the notebook so you can read and execute the code it contains.
2. Read the instructions at the top of the notebook, and execute the cells as instructed. Remember you can use SHIFT + ENTER to execute the currently selected cell, and if you do not have a cluster attached, you will be prompted to attach to the cluster you recently deployed.

## Exercise 4: Completing the solution

Duration: 45 minutes

In this exercise, you will perform the final integration with the Computer Vision API and the Text Analytics API along with the Azure Machine Learning service you previously deployed, to deliver the completed proof of concept solution.

### Task 1: Deploy the Computer Vision API

1. Navigate to the Azure Portal in your browser.
2. Select **Create a resource**.
3. Select **AI + Machine Learning** and then **Computer Vision**.  
   [](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image19.png)
4. On the **Create** blade, provide the following:

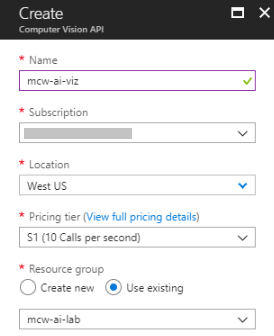
a. **Name:** Provide a unique name for this instance.

b. **Subscription:** Select your Azure subscription.

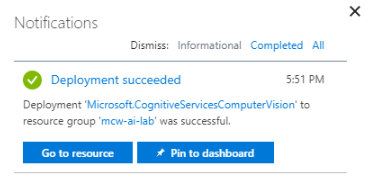
c. **Location**: Select a location nearest your other deployed services.

d. **Pricing tier**: Select S1.

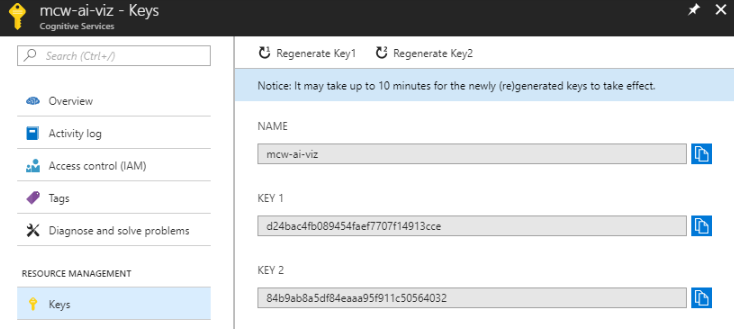
e. **Resource group**: Select the existing mcwailab resource group.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image60.png)

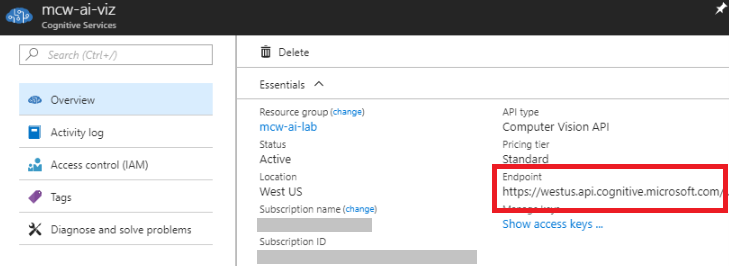
1. Select **Create**.
2. When the notification appears that the deployment succeeded, select **Go to resource**.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image61.png)

1. Select **Keys** and then copy the value of **Key 1** into notepad or something similar as you will need this value later in the lab.

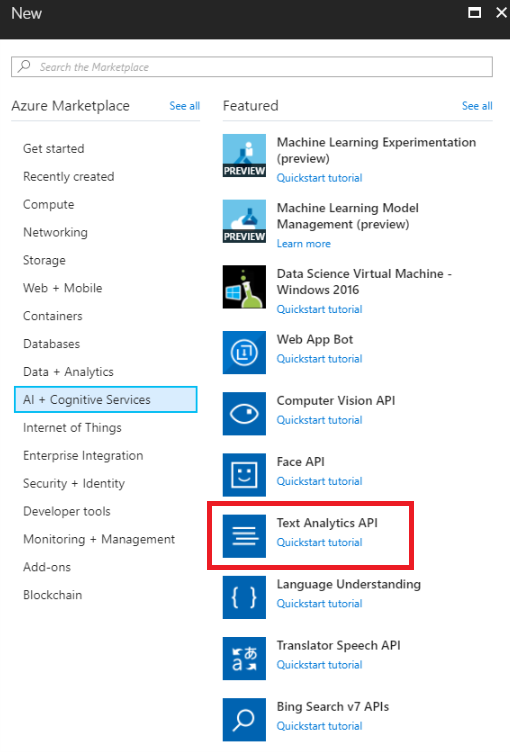
[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image62.png)

1. Select **Overview** and copy the value of Endpoint from the Essentials panel. Store this value in notepad or something similar as you will need this value later in the lab.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image63.png)

### Task 2: Deploy the Text Analytics API

1. Navigate to the Azure Portal in your browser.
2. Select **Create a resource**.
3. Select **AI + Machine Learning** and then **Text Analytics**.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image64.png)

1. On the **Create** blade, provide the following:

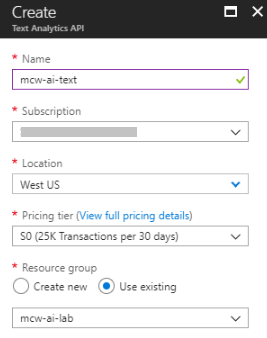
a. **Name**: Provide a unique name for this instance.

b. **Subscription**: Select your Azure subscription.

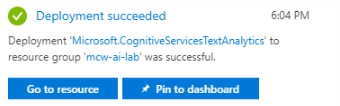
c. **Location**: Select a location nearest your other deployed services.

d. **Pricing tier**: Select S0.

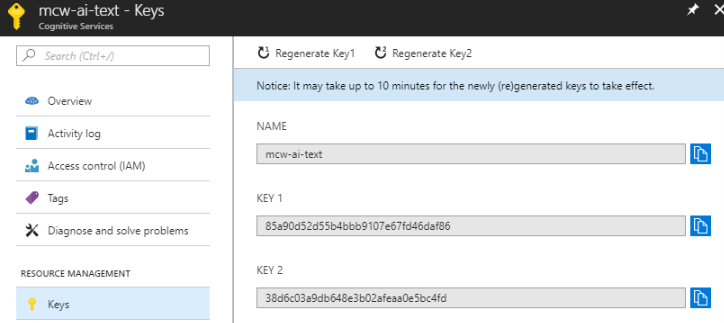
e. **Resource group**: Select the existing mcw-ai-lab resource group.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image65.png)

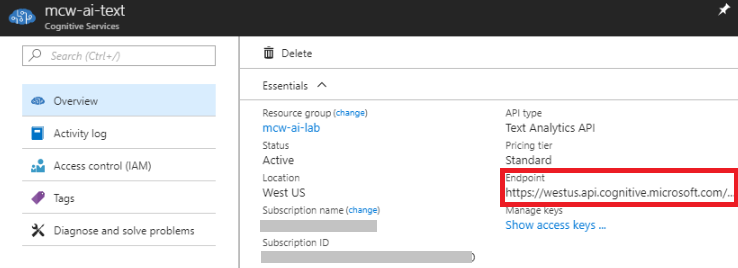
1. Select **Create**.
2. When the notification appears that the deployment succeeded, select **Go to resource**.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image66.png)

1. Select **Keys**, and then copy the value of **Key 1** into notepad or something similar as you will need this value later in the lab.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image67.png)

1. Select **Overview** and copy the value of Endpoint from the Essentials panel. Store this value in notepad or something similar as you will need this value later in the lab.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image68.png)

### Task 3: Completing the solution

1. Return to your Azure Databricks Workspace. Select the Workspace item in the menu and navigate to the folder where you uploaded the Databricks Archive (which should be [your-name/AI-lab]), and select the notebook called 05 Cognitive Services. Follow the steps within the notebook to complete the lab and view the result of combining Cognitive Services with your Azure Machine Learning Services.

## Lab 06: Develop Power BI Report to Surface Analysis

TODO: Lab overview

Prerequisites

* Previous labs completed successfully

Exercise 1: TODO: Title

TODO: Exercise overview

*Task 1: TODO: Title*

1. TODO: Steps

## After the hands-on labs

Duration: 5 minutes

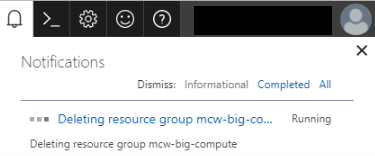
To avoid unexpected charges, it is recommended that you clean up all your lab resources when you complete the lab.

### Task 1: Clean up lab resources

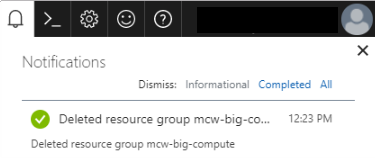
1. Navigate to the Azure Portal and locate the **MS-DIRW** Resource Group you created for this lab.
2. Select **Delete resource group** from the command bar.

[Screenshot of the Delete resource group button.](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image71.png)

1. In the confirmation dialog that appears, enter the name of the resource group and select **Delete**.
2. Wait for the confirmation that the Resource Group has been successfully deleted. If you don't wait, and the delete fails for some reason, you may be left with resources running that were not expected. You can monitor using the Notifications dialog, which is accessible from the Alarm icon.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image72.png)

1. When the Notification indicates success, the cleanup is complete.

[](https://github.com/Microsoft/MCW-Cognitive-services-and-deep-learning/blob/master/Hands-on%20lab/media/image73.png)