# Create a Data Solution on Azure Synapse Analytics with Snapshot Serengeti: A step by step guide for beginners.

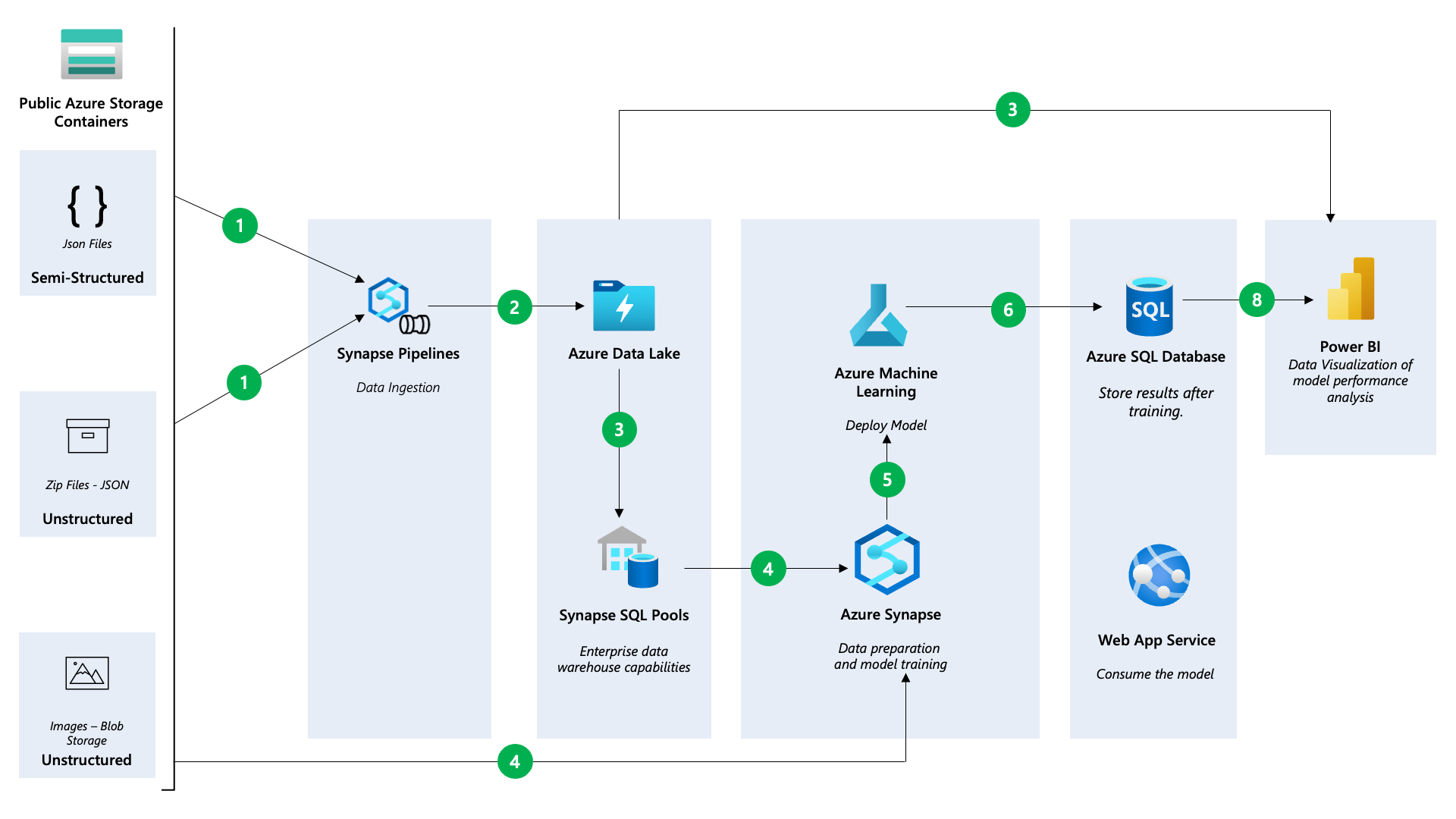
This will be the first article in a four-part series on building an end-to-end data analytics and machine learning solution on azure synapse analytics.

The data used in this solution is the Snapshot Serengeti data set which is a large-scale dataset of camera trap images. To learn more about this dataset see https://lila.science/datasets/snapshot-serengeti.

By the end of this series, you’ll learn how to transform and analyze this data then use it to train classification ML model to classify the camera trap images to one of the 48 species in the dataset.

The solution described in this article combines a range of Azure Services that will ingest, store, transform and serve data and insights from different sources. i.e,

1. Semi-structured data - these are Json files that contain the split of the dataset into train, test and validation sets.
2. Unstructured data - these are the images in the dataset and also the zipped files that contain the metadata of the images.

* 

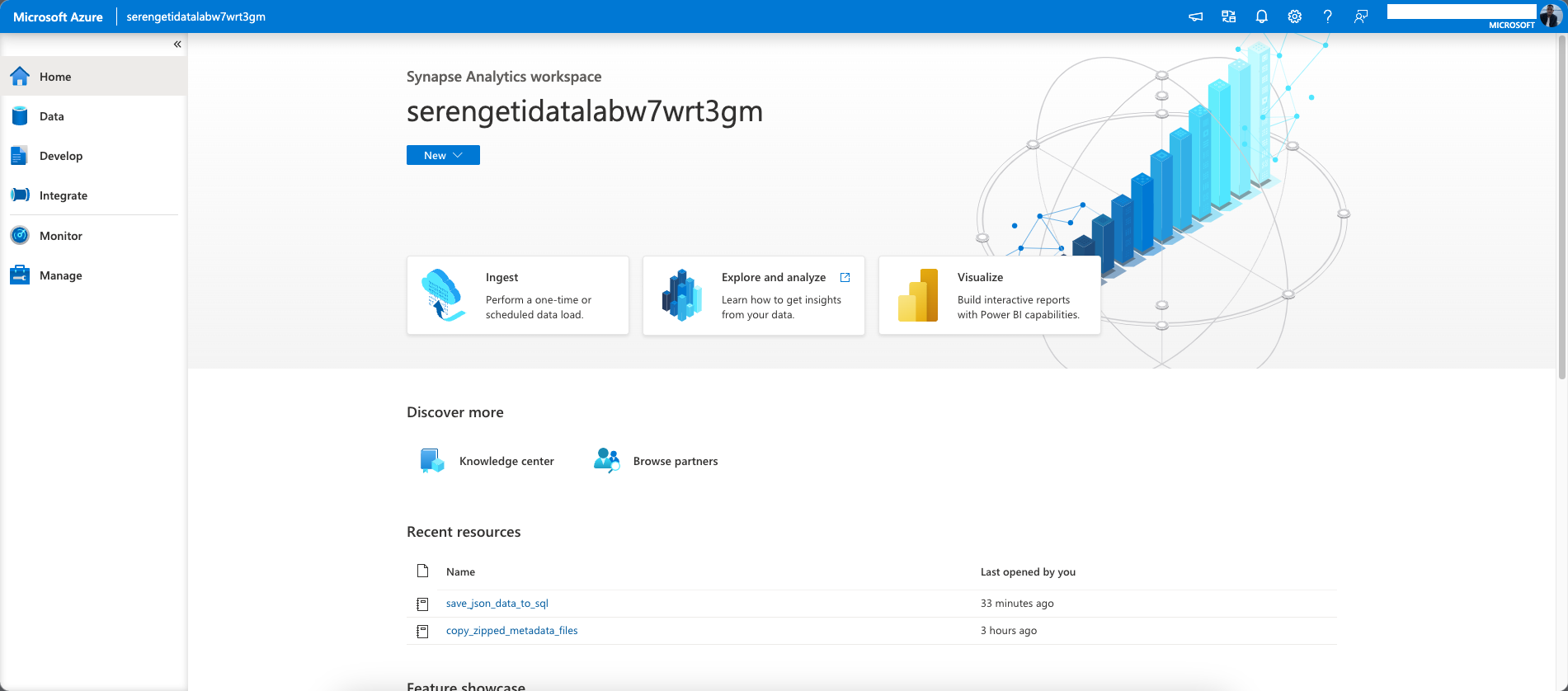
There are different ways to building this data solution but we’ll go with one that allows you to interact with most of the services on Azure synapse. Before proceeding check the set up instructions as explained in this GitHub repository: https://github.com/Jcardif/serengetidatalab on how to deploy the resources needed for this solution to azure.

Now that you have successfully deployed the resources needed for this solution, the remainder of this article will look at how we load the data into the [storage account](https://azure.microsoft.com/en-gb/free/storage/?WT.mc_id=data-89327-jndemenge), [transform the data](https://learn.microsoft.com/en-US/Azure/synapse-analytics/synapse-notebook-activity?WT.mc_id=data-89327-jndemenge&tabs=classical) and finally load it into the [SQL Data warehouse](https://learn.microsoft.com/en-us/azure/synapse-analytics/get-started-analyze-sql-pool?WT.mc_id=data-89327-jndemenge).

## Synapse Analytics Workspace

From the deployed resource group open the Synapse Workspace resource then click on **Open Synapse Studio**. This will open the Synapse Studio in a new tab. Synapse studio allows you to execute all data tasks with a simple UI and unified environment.

Synapse studio is divided into Activity Hubs, which organize the tasks needed for building analytics solutions. These hubs are: 1. Data Hub - Explore data inside the workspace and in linked storage accounts 2. Develop Hub - Develop and run notebooks, data flows and SQL scripts 3. Integrate Hub - Create and manage data pipelines 4. Monitor Hub - Monitor the health of the workspace and the pipelines 5. Manage Hub - Manage the workspace, configurations and linked services



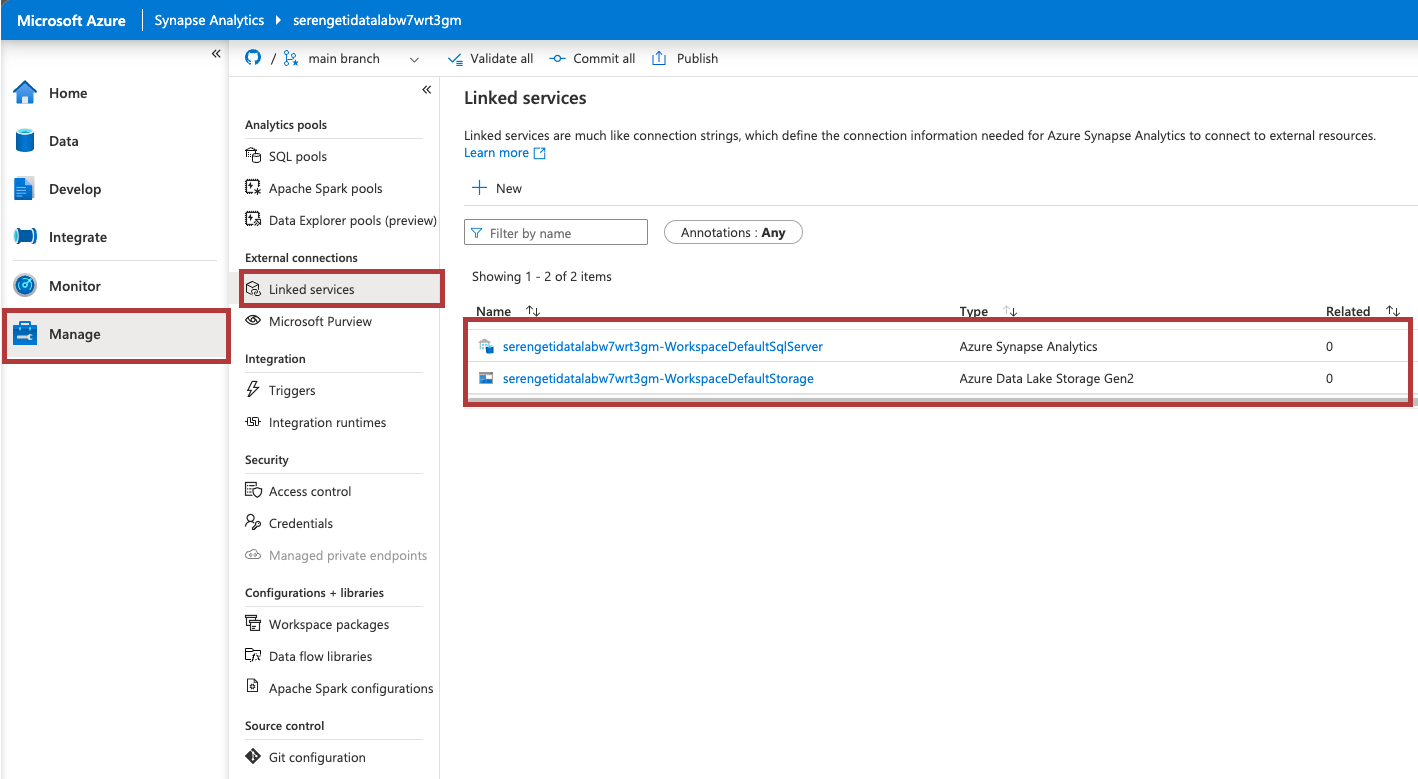
## Managing workspace configurations & Linked Services

Navigate to the **Manage** hub. This hub allows you to manage SQL Pools, Apache Spark Pools and Linked Services as well as manage workspace configurations and access.

SQL Pools are fully managed SQL Server instances that can be used to run SQL scripts and queries. Apache Spark Pools on the other hand provide big data compute capabilities in the workspace. Both of these are key components of a big data solution where the SQL Pools provide the data warehousing component and the Spark Pools provide the compute capabilities.

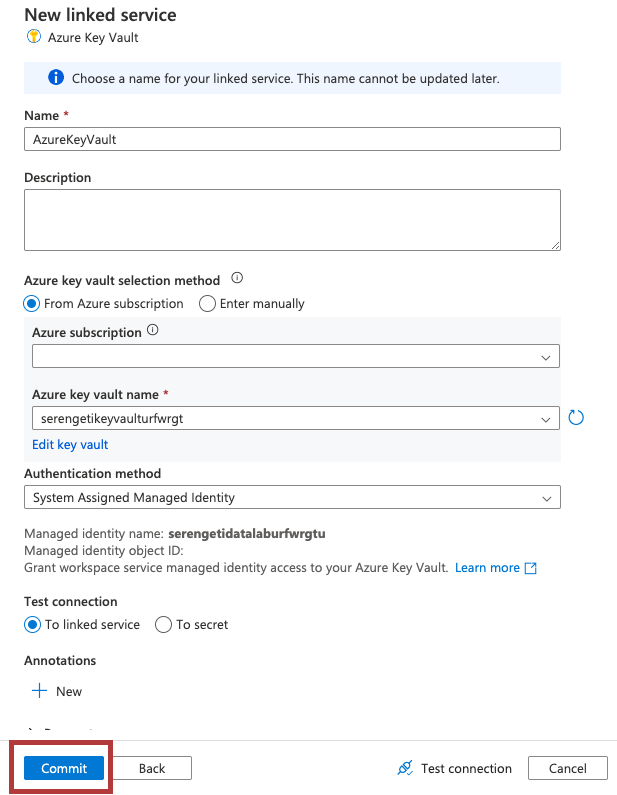
Navigate to the **External connections** section, and click on **Linked services**. This opens a list of the linked Services in the workspace. Linked services in Synapse are used to connect to external resources.

By default you there are two linked services to the he Dedicated SQL Data warehouse and the Azure Data Lake Storage Gen which provides the storage component for your data solution on azure.



Let’s create a new **Linked Service**. In this case we’ll be linking Azure Key Vault to the Synapse workspace. This will allow us to retrieve stored credentials from the key vault.

Click on **New** and search for **Key Vault**. Select the **Key Vault** option and click on **Continue**. In the dialog that appears, provide a name for the linked service, select your subscription and the key vault name then commit.



After successfully creating the linked service, publish the changes to the workspace by clicking on the **Publish** button at the top of the workspace.

Now that we have successfully linked to Azure KeyVault we can begin data ingestion, leveraging the Notebooks in synapse to load the data into the storage account.

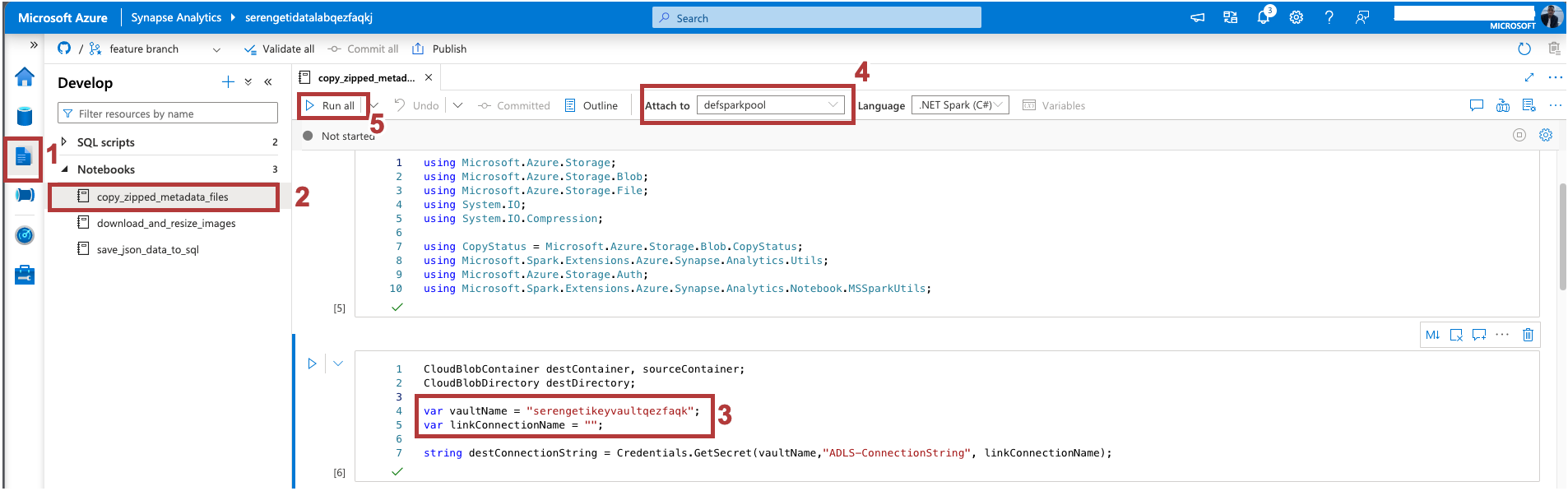
## Ingesting the data - Synapse Notebooks, Spark and Azure Data Lake Storage Gen2

Synapse Notebooks are a great way to interact with data in the workspace. They allow you to write code in different languages to interact with data in the workspace. In this case we’ll be using C# notebooks to load the data into the storage account.

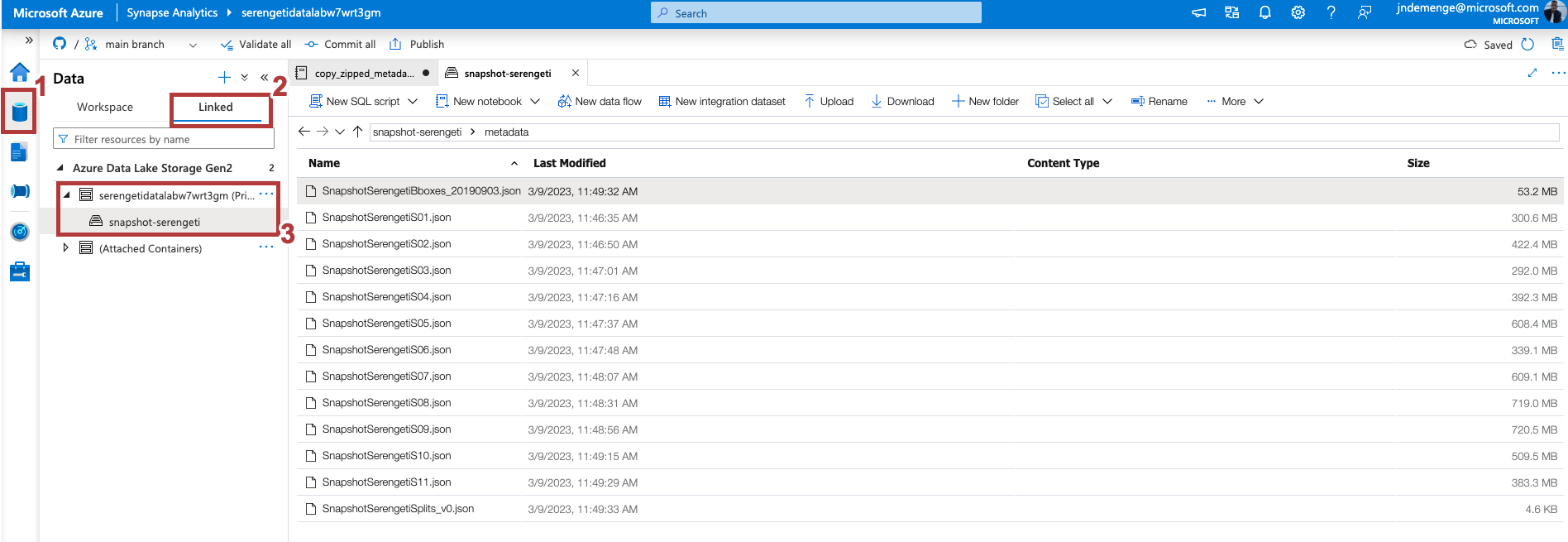
To begin, navigate to the **Develop** hub and expand the **Notebooks** and you’ll find three notebooks, i.e.: 1. copy\_zipped\_metadata\_files - this notebook downloads and extracts the metadata files from the Snapshot Serengeti dataset into the storage account. 2. save\_json\_data\_to\_sql - this notebook loads the json files into the SQL data warehouse. 3. download\_and\_resize\_images - this notebook performs image transformation before using the image data for training the ML model.

To begin, open the **copy\_zipped\_metadata\_files** notebook and in the third cell replace the values of the vaultName and linkConnectionName variables with the name of the key vault and the linked service name to the key vault respectively.

Next attach the notebook to a spark pool by clicking on the **Attach to** button at the top of the notebook. Select the defsparkpool, next click on the **Run all** button to execute the notebook.



Navigate to the **Data** hub and clicked the **Linked** tab, expand the **Azure Data Lake Storage Gen2** and the AZure Data Lake storage account and you’ll see a newly created container called **snapshot-serengeti**. Open the container and inside the metadata folder you’ll find the extracted metadata files.

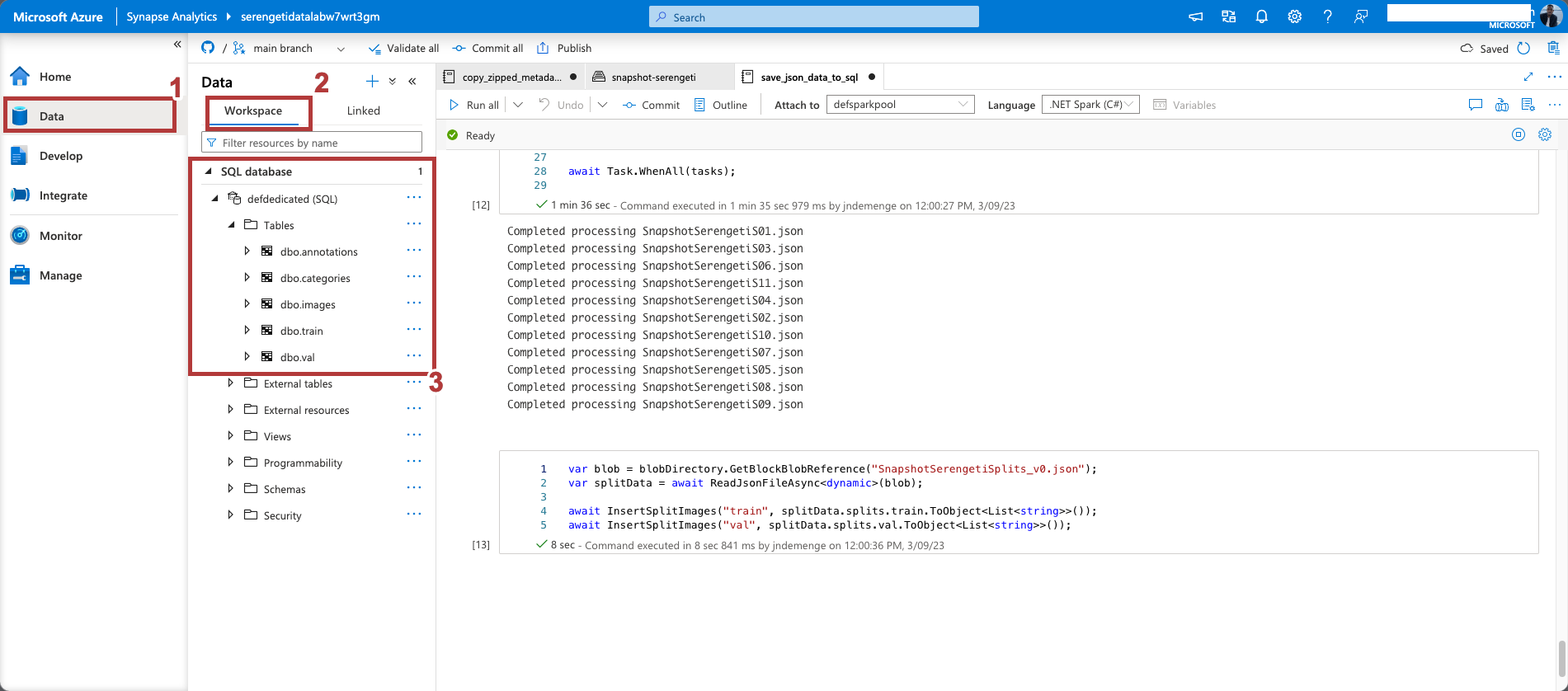


## Loading & Transforming the data - Synapse Notebooks, Spark and SQL Data Warehouse

Open the **save\_json\_data\_to\_sql** notebook and in the fourth cell replace the values of the vaultName and linkConnectionName variables with the name of the key vault and the linked service name to the key vault respectively.

Attach the notebook to the defsparkpool and run the notebook. This will load the data from json files into the SQL data warehouse.

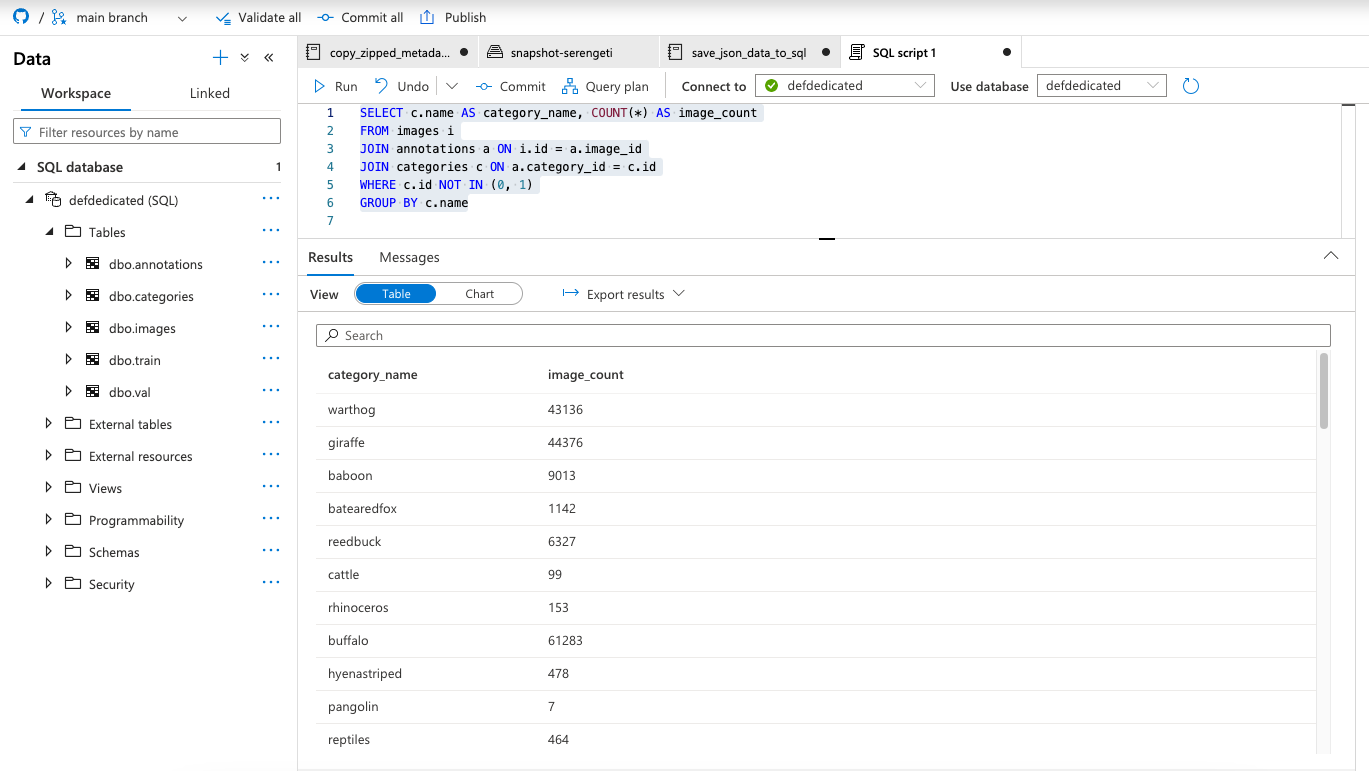
After successful execution, navigate to the **Data** Hub and in the **Workspace** tab expand the **SQL databases** and the **defdedicated** database and inside the **Tables** folder you’ll find the tables created from the json files.



Right click on the **defdedicated** database and select **New SQL script** to open the SQL script editor. In the editor, run the following query.

SELECT c.name AS category\_name, COUNT(\*) AS image\_count  
 FROM images i  
 JOIN annotations a ON i.id = a.image\_id  
 JOIN categories c ON a.category\_id = c.id  
 WHERE c.id NOT IN (0, 1)  
 GROUP BY c.name

This query gets the number of animals in each category excluding the empty and human categories.



## Conclusion

In this article we explored how you can leverage Azure Synapse Analytics for your big data solutions, highlighting some the Key components and how you can link to external resources.

In the next article we’ll explore how we connect to power bi for data visualization and how we can link to Azure Machine Learning for training and deploying machine learning models.

## Resources

For additional resources to get an in-depth understanding of the services discussed in this article take a look at this handy collection of resources:

* Azure Synapse Analytics Blog - https://aka.ms/SynapseBlog/
* Documentation - http://aka.ms/SynapseDocs
* Learning paths and modules - https://aka.ms/synapsemodules
* Azure Synapse YouTube videos - https://aka.ms/SynapseYouTube
* Microsoft Power bi Community - http://aka.ms/pbicom

## Deploying your own infrastructure?

To follow along in this demo, you deployed the infrastructure using the ARM template provided in the GitHub repository. Learn how you can use [Bicep](https://learn.microsoft.com/en-us/azure/azure-resource-manager/bicep/learn-bicep?WT.mc_id=data-89327-jndemenge) to build and deploy your infrastructure as code.

New to this? [GitHub Copilot](https://docs.github.com/en/copilot/quickstart) can assist you.

## Publish

published at https://aka.ms/synapseserengeti