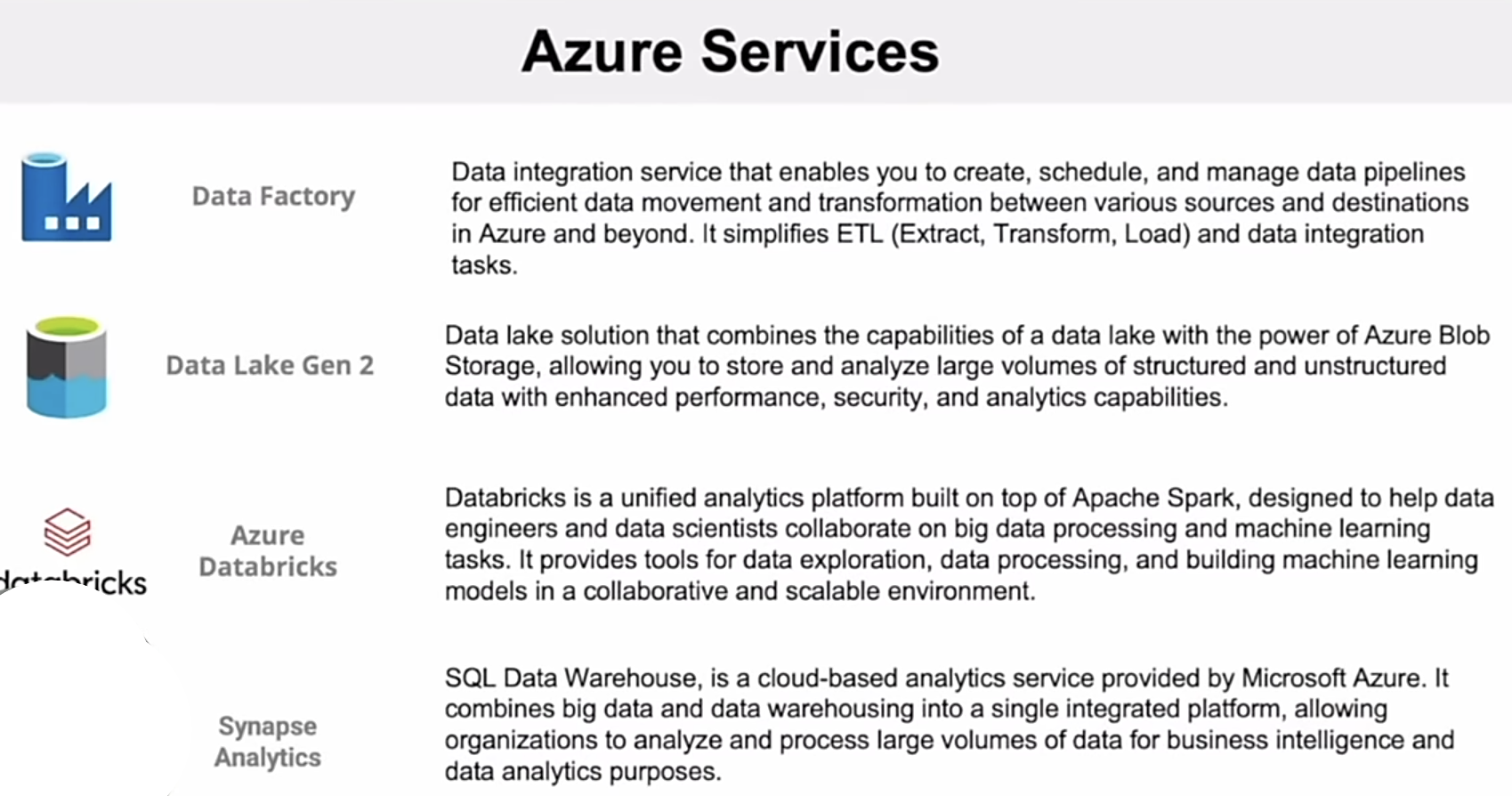
Une image contenant texte, diagramme, carte, capture d’écran

Description générée automatiquement

* Create an account with email (First Level)
* With that account, possibility to create multiple subscriptions (Free trial, eng. subscription, …). So based on the budget, you can allocate different subscription plans to different departments.
* Inside a subscription, you have a resource group (logical grouping of different resources such as data factory, data lakes, apps analytics, Azure database, …)



* Azure blob: the object storage. When you store a file on Azure blob, it is considered as an entire object (so inside a bucket, each file is considered as an object)

**Create an account**

* Azure cloud > First link

<https://azure.microsoft.com/fr-fr/>

* Click on Free Account / Start with Azure > Try Azure freely
* Create a Microsoft Account (if you don’t have yet)
* Create your Azure profile (Make sure you have a credit card)
* Access to the Azure portail

<https://portal.azure.com/>

Une image contenant texte, conception, capture d’écran

Description générée automatiquement

**Set up our data lake (e.g. folder in our container ???)**

* Create a storage account: will allows you to store object data
* Create a new resource group: tokio-olympic
* Give a name to the storage account: should be unique in all Azure worldwide (e.g.: tokyoolympicdatavad)
* Select the nearest region for example
* Choose the performance
* Specify redundance: if you want for e.g. to replicate the data across different data centers or region
* Click on next
* Activate the hierarchical Namespace: all data stored will be available in a hierarchical format (as you could have seen it on your computer and not as an object)
* Click on next
* Click on next
* Click on next
* After the validation, click on “Create”
* Click on “Go to resources”
* At the left window, we are interested in “data storage” > “containers”: store the data as an object
* Click on “containers” > “create container” > Give a name (e.g.: tokyo-olympic-data) > “create”
* Click on our container “tokyo-olympic-data” > “add directory” > Give a name (e.g.: raw-data) > “save”
* Create another directory > Give a name (e.g.: transformed-data) > “save”

**Use Data Factory for data ingestion (source to raw-data)**

* In the search bar, search for “Data factories” > Open it in a new tab > “Create data factory”
* Select our resource group (tokio-olympic) > Give a name (e.g. tokyo-data-dt)
* Select region and version > Next until the end > Create > “Go to resources” > “Launch Studio”

Left panel:

* Home: where all data factors will be showed

Une image contenant texte, Police, Bleu électrique, nombre

Description générée automatiquement

* Author: where we create the pipeline

Une image contenant texte, Police, nombre, ligne

Description générée automatiquement

* Monitor: monitor each part of the pipeline (error, …)

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

* Manage: configure things

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**Building the data ingestion pipeline**

* Click on Author > “+” button > Pipeline > Pipeline > Right side, Give a name: (e.g.: data-ingestion)

Left side: we have the available activities (these are operations we want to perform): many activities available that we can integrate in our data factory

* Click on “Move and transform” > Drag and drop “Copy Data” (to copy data from the source to our data storage) > General (Give a name: Athletes) > Below “Source”

Go to Github, choose any of the csv file > Click on raw > Copy the URL > Create a Link in data (Below “Source” > “New” > Search for “HTTP” > Continue > Specify File Format > Continue > Give a name: “Athletes” > New > Give a name: AthletesHTTP > Base URL: all the url > Authentication type : Anonymous > Create > OK > Preview data (below)) > Past it in “Source”

<https://raw.githubusercontent.com/avadombi/DataEng/refs/heads/main/001-Azure/001_Projet1/dataset/Athletes.csv>

* Below “Sink” > New > Azure Data Lake Storage Gen2 > Continue > Select the storage file’s type (e.g. CSV) > Continue > Give a name: ADLS (Azure Data Lake) > New > Account selection method > Azure subscription > storage account : tokyoolympicdatavad > Create > Click on the “directory” icon on the side > tokyo-olynpic-data > raw-data > OK > Specify a file name: **athletes.csv > import schema: None > OK**
* Click on “Validate” (above) > Close > Debug (above): run the code to test if it run or not
* Redo for the other data files

**Build the data transformation pipeline using Databricks**

* Search for “Azure databricks” > Create Azure Databricks Service > Specify subscription > Select resource group > Workspace name: tokyo-olympic-db > Pricing Tier : Premium (+ Role-based…) > > Next: Networking until the end > Review + Create > Create > “Go to resources” > “Launch Workspace” > Automatic Sign in

**Create Azure databricks workspace**

* Left side, click on “Compute” > Right side: click on “Create compute” > Policy: Unrestricted > Select Single node (possibly to use multiple computers or workers at scale but here not) > runtime: default one > Node type: default one > Click on “Create compute” > Wait and at the end, click on “Compute again”

**Get the data from Azure Data Lake, transform it and store the result in ADL**

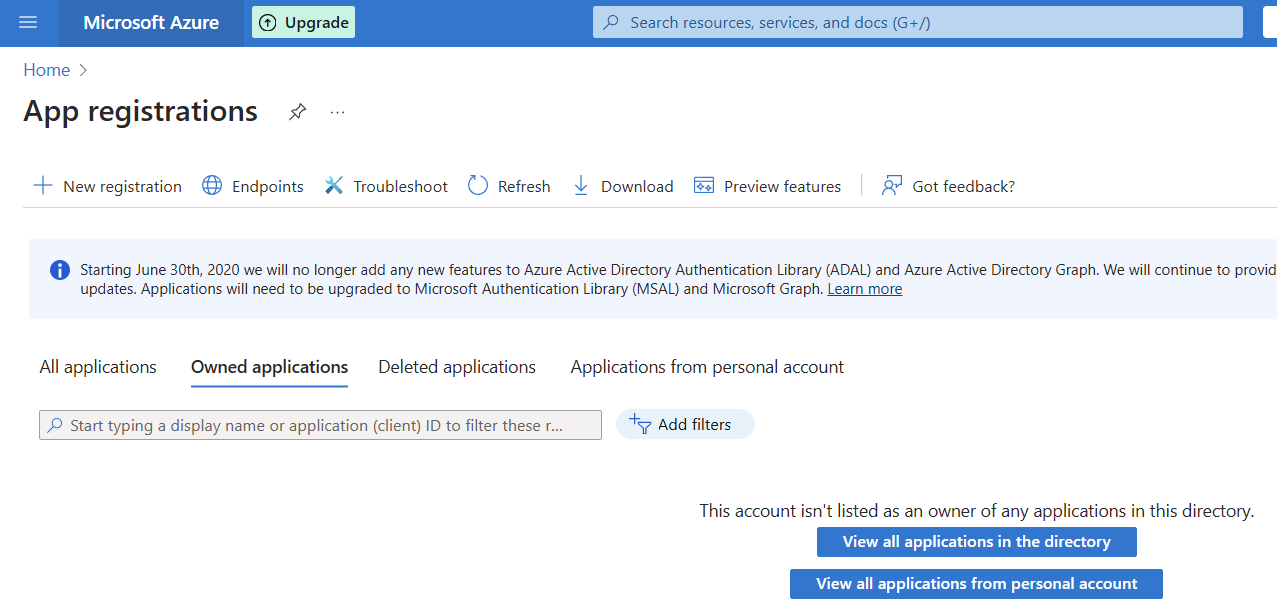
* Click on “New” > Notebook
* Rename the notebook: Tokyo Olympic Transformation
* Create the connection to ADL: mount the ADL storage (as in Colab)

First, select our Spark Cluster

Une image contenant texte, capture d’écran, Police, Logiciel multimédia

Description générée automatiquement

Go to Azure Home page and search for “App registrations”: will allow to create the connection



Click on “New registration” > Give a name: app01 > Register

Application (client) ID: c21eb02e-fa8b-46e9-8071-ada73981c566

Directory (tenant) ID: 2a52e8a9-349d-4492-9580-cf202e778a20

Left, side: Click on “Manage” > Certificate & secrets > Click on “New client secret”: Description => (eg. secretkey) > Click on “Add” > Copy the value of “secretkey”:

Secretkey: -i18Q~qJSnDkUOZ9Q2kZdFUr\_pg9tMGBJ1csibKk

**Create connection to Azure Data Factory**

* Create the configuration format in Databrick

Une image contenant capture d’écran, texte, Police

Description générée automatiquement

This not the best practices when working in a real word to expose theses id in the code. Instead, use “Keys” from the search bar of “Azure Home Page”

Go to the Azure Home Page and search: storage account (**Section 1**)

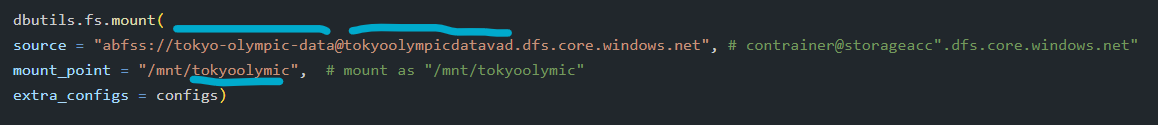
Une image contenant texte, capture d’écran, Police, diagramme

Description générée automatiquement

Click on “tokyoolympicdatavad” (storage account name)

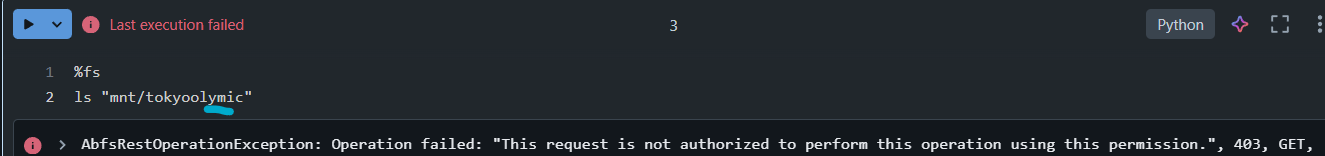
Click on Container and copy “tokyo-olympic-data” (container name)

(Below the above code)



Run

To ckeck that we have successfully mounted, let’s list our folders:



Go to give permission to access:

* Go on your storage (as in **Section 1**) > Click on Containers > tokyo-olympic-data > Left Side: Click on access Control (IAM) > Add > Add Role Assignment (Above) > Go below and click on “Storage Blob Data Contributor” (give access to read, write, …) > Next > Select members > Write: **app01** > Click on it > Select > Next > Next > Review + assign

**Re-Run**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Now, let’s load a data (e.g.: athletes.csv)

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

**Do the same for all**

Une image contenant capture d’écran, texte, Police

Description générée automatiquement

**Look at data scheme**

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

**Basic Data transformation**

Documentation Link for transformation functions:

<https://spark.apache.org/docs/latest/rdd-programming-guide.html>

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Une image contenant texte, logiciel, Logiciel multimédia, Police

Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Store the transformed data**

Une image contenant texte, logiciel, Logiciel multimédia, capture d’écran

Description générée automatiquement

Repartition(n): will partition data into “n” files