**DSBA 6190 | Data Platform Lab**

**Description:** In this lab, we’ll focus on the creation of a data lake for us to house some data. Using both the Azure Portal and the Azure CLI, you’ll learn how to create an Azure Storage Account with the appropriate settings to turn it into a data lake. Plus, you’ll learn how to create containers and upload data.

**Notes:**

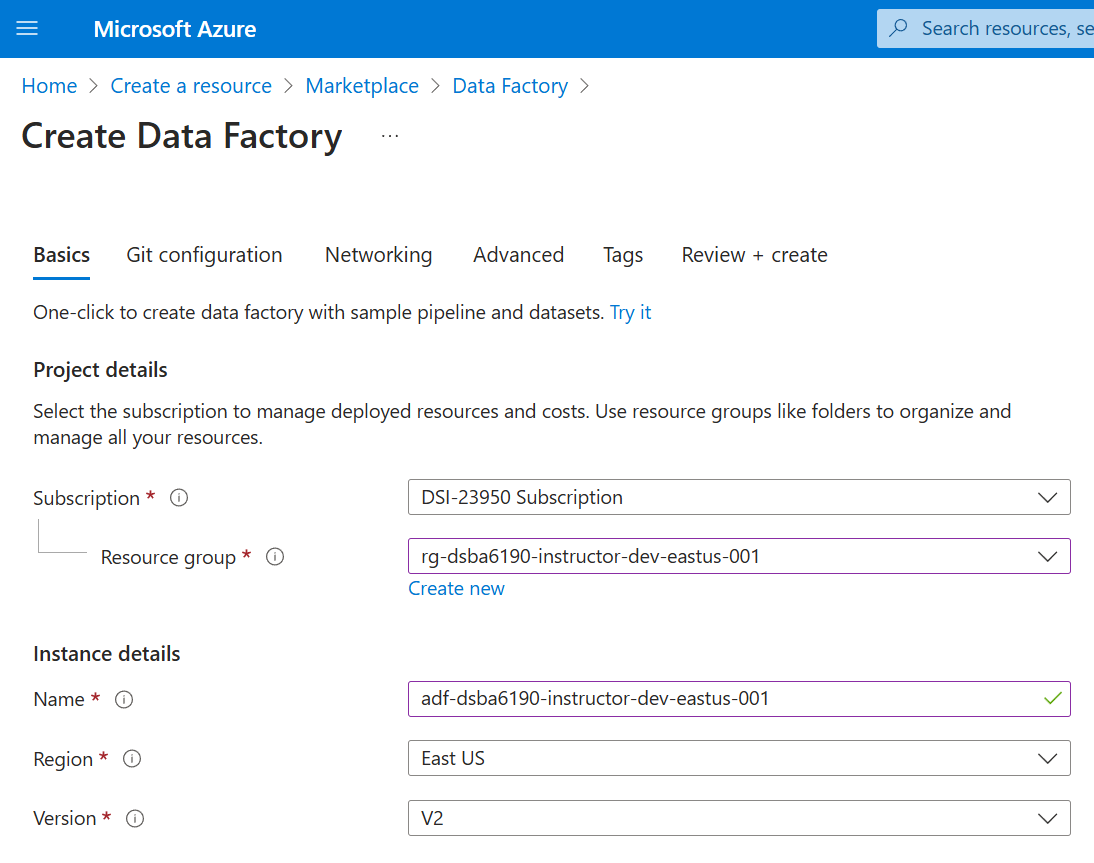
* Only provision the requested resources using the defined settings. Remember, the class cloud budget is everyone’s responsibility.
* Use the standard [Azure naming conventions](https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming) when your name your Resource Group and any services you create.

## Steps:

1. Create an Azure Data Factory. (ONLY 1 PER GROUP)
2. Create an Azure Storage Account. (ONLY 1 PER GROUP)
3. Create a “data” container from the Azure Portal. (ONLY 1 PER GROUP)
4. Login to your Azure account using the Azure CLI. (EACH GROUP MEMBER)
5. Upload a sample file to your folder in the “data” container using the CLI/SDK. (EACH GROUP MEMBER)
6. Copy a sample file to your folder in the “data” container using Data Factory. (EACH GROUP MEMBER)

# Step 1: Create the Azure Data Factory

First, create a Data Factory for your class group.



On the **Create Data Factory** page (**Basics** tab), use the following:

* Subscription: DSI-23950 Subscription
* Resource group: rg-dsba6190-<GROUP NAME>-dev-eastus-001
* Name: adf-dsba6190-<GROUP NAME>-dev-eastus-001
* Region: East US
* Version: V2

Leave the default settings on the **Git configuration**, **Networking**, and **Advanced** tabs.

A screenshot of a computer

Description automatically generated

On the **Tags** tab, add the following tags:

* class: dsba6190
* semester: fall2024
* instructor: cford38
* group: <GROUP NAME>

# Step 2: Create an Azure Storage Account

Next, create an Azure Storage Account for your group.

A screenshot of a computer

Description automatically generated

Settings (under the **Basics** tab):

* Subscription: DSI-23950 Subscription
* Resource group: rg-dsba6190-<GROUP NAME>-dev-eastus-001
* Storage account name: stodsba6190<GROUP NAME> (no spaces, fewer than 24 characters)
* Region: East US
* Primary service: Azure Blob Storage or Azure Data Lake Storage Gen 2
* Primary workload: (Take a look at the options and pick whichever seems to make sense)
* Performance: Standard
* Redundancy: Locally-redundant storage (LRS)

On the other tabs, review the options, but most of the defaults should be fine.

Also, make sure you selected the option that makes this an Azure Data Lake Storage service.

On the **Tags** tab, add the same tags as before.

# Step 3: Create a “data” container

Go to the Storage Account that was just created and create a new container called “data”.

A screenshot of a computer

Description automatically generated

(Hint: If you don’t see the same Containers icon as in the screenshot below, you didn’t create a data lake. You just made a blob storage account.)

A screenshot of a computer

Description automatically generated

# Step 4: Login using the Azure CLI

To interact with the Azure environment from your local laptop, you can use the Azure CLI to login and perform operations.

On your laptop, go to your Command Prompt (on Windows) or Terminal (on macOS or Linux) and type: az login

This will open a browser window where you can sign into the Azure Portal. If successful, your terminal will show the Subscriptions to which you have access.

A screenshot of a computer

Description automatically generated

You can set the default subscription using: az account set --subscription "DSI-23950 Subscription"

# Step 5: Upload a File using the CLI or SDK

Now that you’ve logged in, let’s upload a file to your “data” container in the data lake.

(Find any file you’re willing to upload. It can be a blank text file or a picture of your cat.)

If you’re comfortable in the command line, upload the file to the data lake using Azure CLI commands.

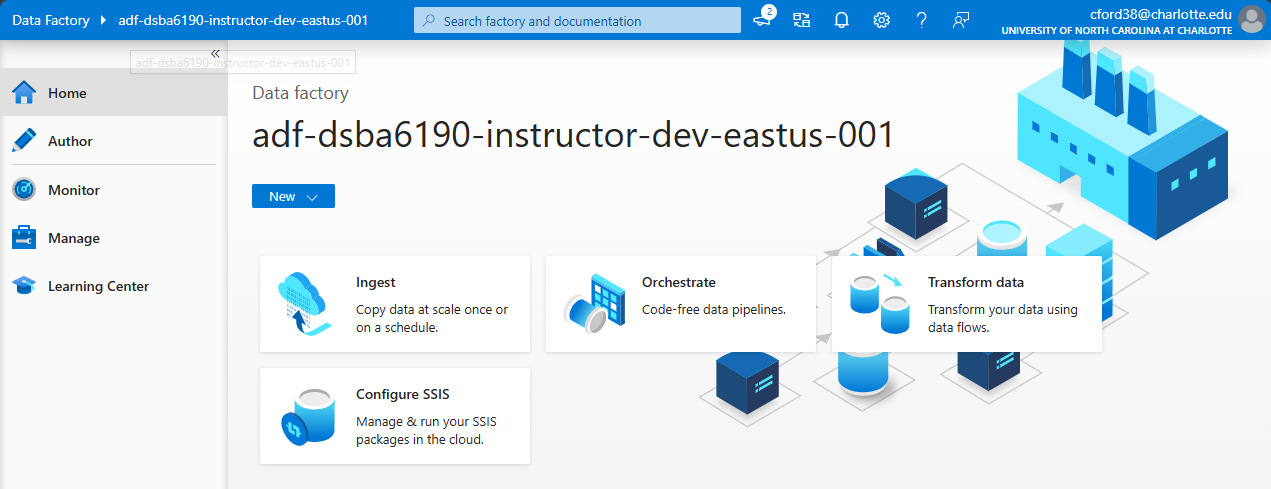
* Documentation: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-cli#upload-a-blob>

If you’re more comfortable in Python, upload the file to the data lake using a Python script.

* Documentation: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-cli#upload-a-blob>

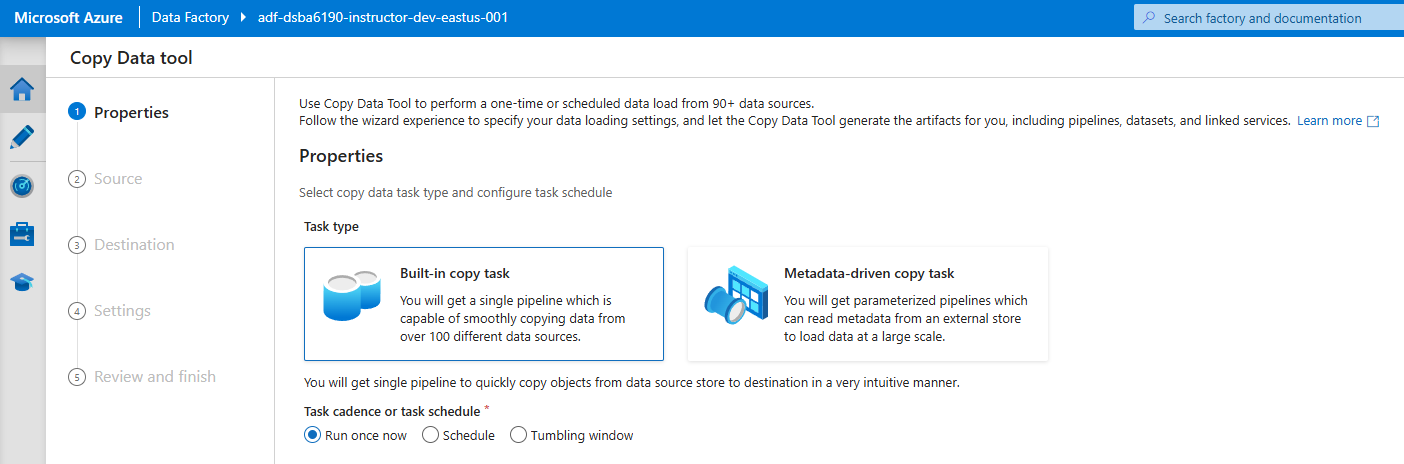
# Step 6: Copy a File using Azure Data Factory

Lastly, create a Data Factory pipeline that will copy da file to your “data” container in the data lake.



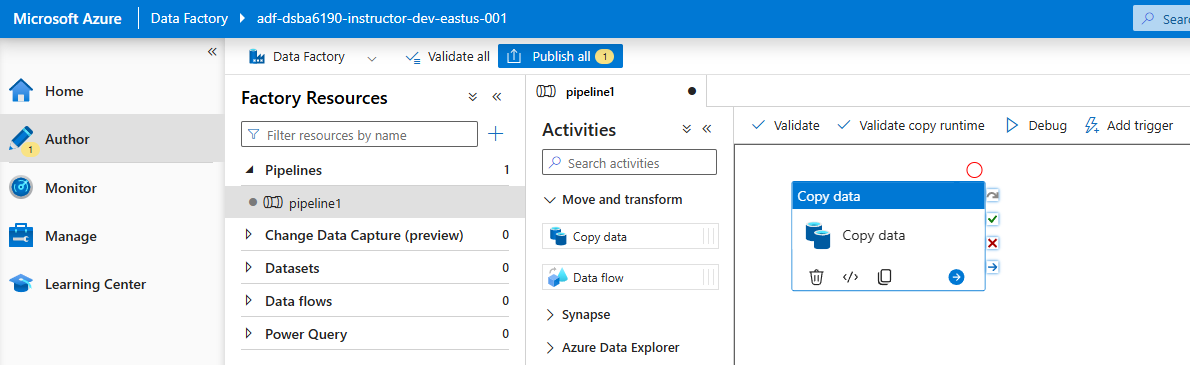
There are a couple ways to copy data using Data Factory.

The easiest is the “Built-in copy task”, which makes a single use pipeline to copy data around.



* Documentation: <https://learn.microsoft.com/en-us/azure/data-factory/quickstart-hello-world-copy-data-tool#use-the-copy-data-tool-to-copy-data>

The more advanced option is to build a pipeline from scratch, which has a lot more options for copying/transforming data.



* Documentation: <https://learn.microsoft.com/en-us/azure/data-factory/load-azure-data-lake-storage-gen2#load-data-into-azure-data-lake-storage-gen2>

# Lab Questions

1. Provide a screenshot of the Storage Account and “data” container that your group made.

A screenshot of a computer

Description automatically generated

1. Using either the Azure CLI or the Azure Storage Python package, list the uploaded files that everyone uploaded to your data lake. (Paste in the command you used and provide a screenshot or the output list.)

A screenshot of a computer

Description automatically generated

1. A screenshot of a computer

   Description automatically generatedProvide a screenshot of your Azure Data Factory pipeline and a screenshot of the data your copied in your data lake. Also, describe the source of your dat

DATA SOURCE: Census B03002 Dataset (Race and Ethnicity)

A screenshot of a computer

Description automatically generated

1. If everyone in your group uploaded 1TB of data each, how much would that cost per month (given the current settings of your data lake)?

We have three people in our group, so chose the 3TB option.

A screenshot of a computer

Description automatically generated

INSERT PHOTO OF LAB

## Understand Database vs. Data Warehouse Differences

#### Questions:

1. If you were asked to create a database or data warehouse for reporting purposes, which would you choose to create and why? The database or data warehouse will have to pull in data from multiple different systems and be focused on getting large amounts of data aggregated for enterprise-level reports.

I would choose a Data Warehouse for a few reasons:

1. Large Data Volumes:
   1. The prompt makes it sound like there will be large swaths of data coming in. A data warehouse would be better served for handling large amounts of data.
2. Multiple Sources:
   1. EDW has better functionality to store data from multiple sources or systems as described in the question.
3. Aggregation of this LARGE data:
   1. Data warehouses again are better supported for handling complex queries such as aggregating data.
4. Data Reporting
   1. We see that databases are usually used for a few fast queries, but data warehouses are typically designed to allow for analytics and reporting as described in prompt.
5. What is/are the difference(s) between a star schema and a snowflake schema in a data warehouse?

Star Schemas are a simpler way to arrange data in a data warehouse. They consist of a fact table in the center connected to dimension tables around it. The fact table consists of measures, while the dimension tables have descriptive attributes.

EX:

* Fact Table
  + Product ID
  + Employee ID
  + Customer ID
    - Dimension Table 1 (Product ID)
      * Product Size
      * Product Color
    - Dimension Table 2 (Employee ID)
      * Employee Name
      * Employee Rank
    - ETC…

Snowflake Schema’s are more complex. They also consist of a central fact table containing measures and dimension tables, but these dimension tables break into more dimension tables. Let’s consider that first example

* Fact Table:
  + Product ID
  + Customer ID
  + Employee ID
    - Dimension Table 1 (Product ID):
      * Product Size ID
      * Product Color ID
      * Product Category ID
        + Dimension Table 2 (Product Size ID)

Product Size ID

Product Size US

Product Size UK

Product Size France

As shown by these examples the Snowflake Schema has a more complex structure, is a bit harder to understand, is a bit slower. However the Snowflake Schema is normalized leading it to be easier to update, require less storage space, and easier to troubleshoot because of this normalization.

## Understand platform selection differences.

#### Questions:

1. What are the capability differences in using Azure DB vs. Azure Synapse?

Azure DB and Azure synapse both allow for cloud-based databases but are utilized for different use cases based on capability differences. The Azure DB allows for a single cloud-based database in which you can build data-driven applications and websites. While the Azure Synapse is an enterprise-level product which includes data warehousing product and the ability to run complex queries across LARGE (petabytes) amounts of data.

Here is a little more specific comparison that we saw in the Data Platform Lecture:

A table with text on it

Description automatically generated

Again, we see that the Azure Synapse is a lot better at those enterprise level applications where a lot of complex queries are going to be happening with larges amount of data. The Azure DB is for smaller relational databases and can support more concurrent queries given that lower amount of data.

1. What are the cost differences in using Azure DB vs. Azure Synapse?

Azure DB costs typically range from $150 - $1,000 per Month while Synapse ranges from $75 - $10,000 (based on 1TB of data). There are a few factors that determine monthly cost per service:

Azure DB Pricing Factors:

* Serverless vs Provisioned
* Redundancy
* vCores

Synapse Pricing Factors:

* Pool Type
* Pool Usage
* Storage
* DWUs

1. What are the capability differences in using Azure Blob Storage vs. Azure Data Lake?

Azure Blob Storage is typically used to store a lot of unstructured data while Data Lakes are used for performing big data analytics.

Data Lakes:

* Allow for Hierarchical File Systems making it easier to manage data in a natural way.
* Allows for Data Analysis with Apache Spark, Azure Databricks, Apache Hive
* Supports two storage options: Hot and Cool
* Increased Security with encryption at REST, Role Based Access Control, but also Access Control Lists which allow you to give permission at different file and folder levels.

Data Blob

* Doesn’t support any data analysis tools or frameworks
* Supports multiple storage options: Hot, Cool, and Archive Options.
* Less Security options with encryption at Rest and Role Based Access Control only.

1. What are the cost differences in using Azure Blob Storage vs. Azure Data Lake?

Azure Blolb Storage and Azure Data Lakes share similar costs depending on the Access Tier and the Redundancy options, but differences start with transaction costs. The Blob Storage options will see lower transaction costs and Data Lakes will see higher transaction costs due to the complexity of the transactions. Data Blobs are intended more as a file storage system, so they will be a bit more cost effective than the Azure Data Lakes where you are doing big data analytics.

1. What was the setting that turned the normal Azure Blob Storage Account into a data lake (Azure Data Lake)?

Enabling the Hierarchical Namespace turns the Blob Storage account into a data lake.

## Industry Use Case

University Supplies Corporation needs to create a data backend for their ordering system. Their system will need to be fast and return data back to the user while placing orders through their website. University Supplies Corporation has locations all across the United States, but their current website and ordering system is hosted on Azure in the West Central US region.

#### Questions:

1. Would you recommend University Supplies Corporation create a database or a data warehouse? Why?

I would recommend that University Supplies Corporation creates a database. They are probably working with transactional data which is better suited for the database which can support real-time updates and has much lower latency than a data warehouse would. A data warehouse is typically better at handling historical data and creating analytics/historical trends which isn’t really what they are looking for in this use case.

1. Would you recommend University Supplies Corporation use Azure SQL DB or Azure Synapse? Why?

I would recommend that University Supplies Corporation use the Azure SQL DB, again this looks like it will be a transactional dataset which the SQL DB will handle better and allow for real-time updates and lower latency; we can see this as SQL DB can process 6400 concurrent queries while the Synapse can only handle 128. Azure Synapse would be better at handling more historical data and creating analytics based off that.

1. Which region would you provision the Azure SQL DB/Synapse? (Bonus: Are there any considerations or capability limitations for choosing this region?)

I would choose either the Western Central US Region or the Central US Region. These are the most geographically centered locations in the US, and this is a national company. This will allow for the best latency for the most number of customers. These regions are a bit more expensive, but the increased speed seems important to the company. The Central region contains an Availability Zone which would add increase protection of the website always being up, so this would be the safest option.

I would either chose the Wester Central US Region or just the Central US Region. There are customers throughout the United States, so having a geographically centered location would be better for optimizing latency for all users; like if we had it in the east region users in the west would have slower performance which isn’t the best. If the business is concerned about resilience the US Central region would be better because it supports Availability Zones while Western Central doesn’t. Note that while West

1. If University Supplies Corporation is expecting to house ~500GB of data in the database/data warehouse, how much would you expect for that service to cost them per month? List all your assumptions like the number of vCores, Billing Option, Backup Option, etc.

Azure SQL Database:

Compute Cost: $531.90

* Region: Central US
* Type: Single Database
* Purchase Model: vCore
* Service Tier: General Purpose
* Computer Tier: Provisioned
* Hardware Type: Standard Series (Gen 5)
* Instance: 4 vCore
* Disaster Recovery: Primary or Geo Replica
* Redundancy: Locally Redundant 1 Database
* Reservation: 3 Year Charge Monthly (we are going full in on this for three years so better be good)
* SQL License: Pay as you go
* Compute Cost: $531.90 (all that stuff above is the calculated in that compute cost)
* Data: 500 GB x 1 Database x $.138 Per GB/Month = $69
* Log: 150 GB x 1 Database x $.138 Per GB/month = $20.70
* Redundancy: RA-GRS
* Point-In-Time Restore: 500 GB x $.240 Per GB/Month = $120
* Retention: 3 Years
* TOTAL COST PER MONTH: $1,000.09