**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:**

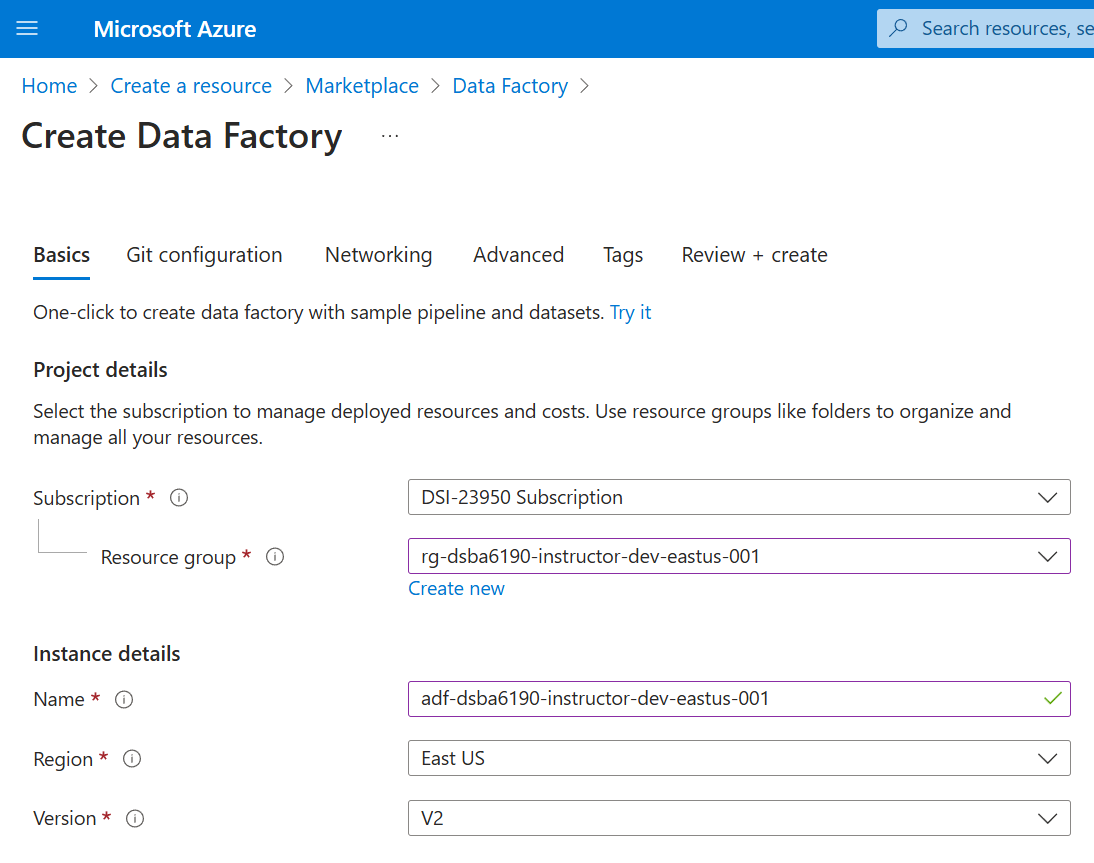
* Only provision the requested resources using the defined settings. Remember, the class cloud budget is everyone’s responsibility.
* Use the standard [Azure naming conventions](https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming) when your name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

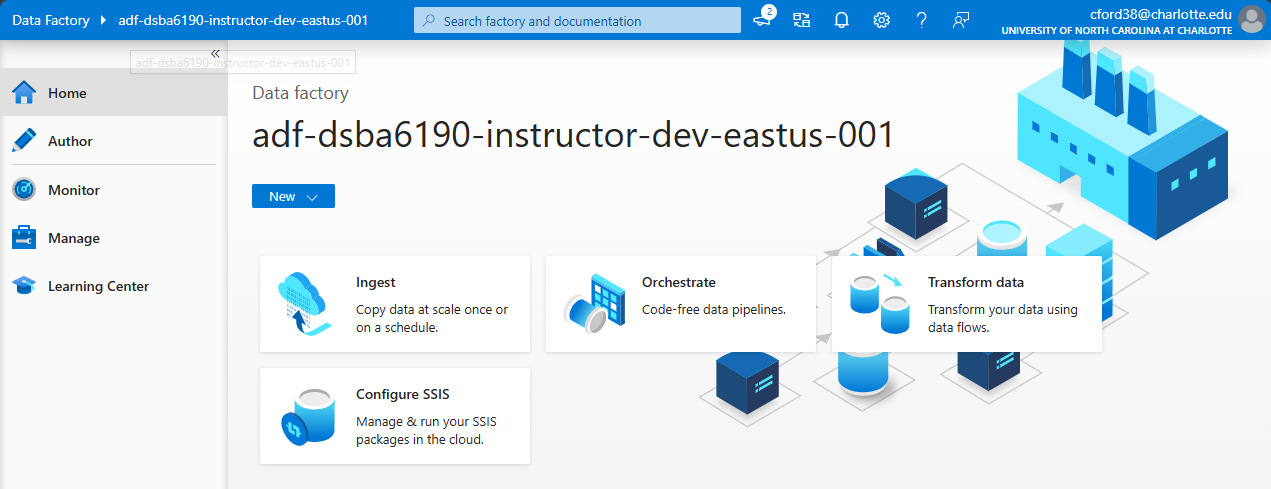
* Documentation: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-cli#upload-a-blob>

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>

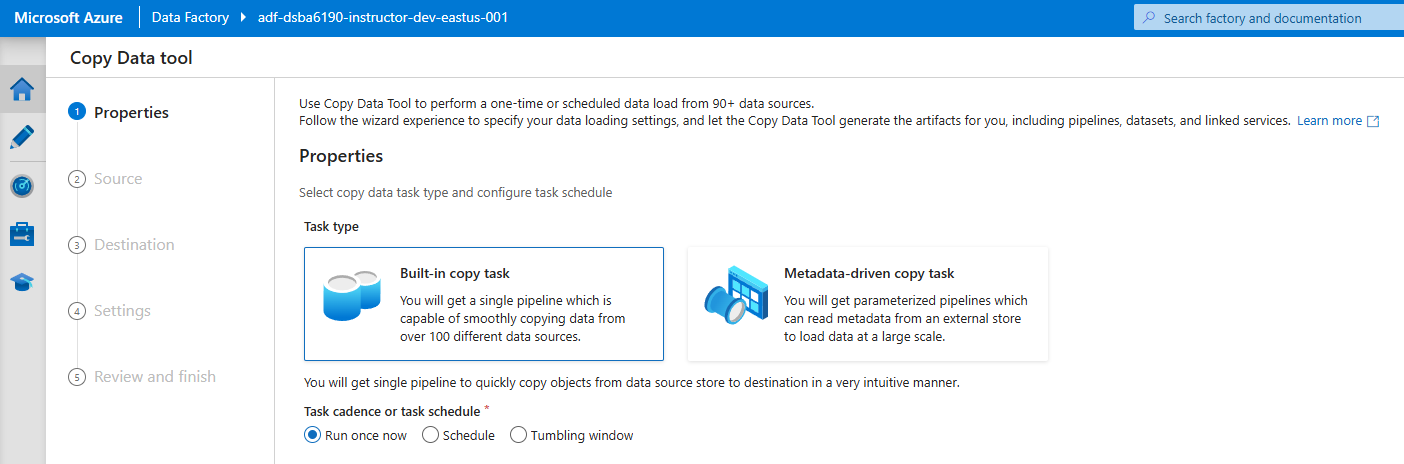
# Step 6: Copy a File using Azure Data Factory

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



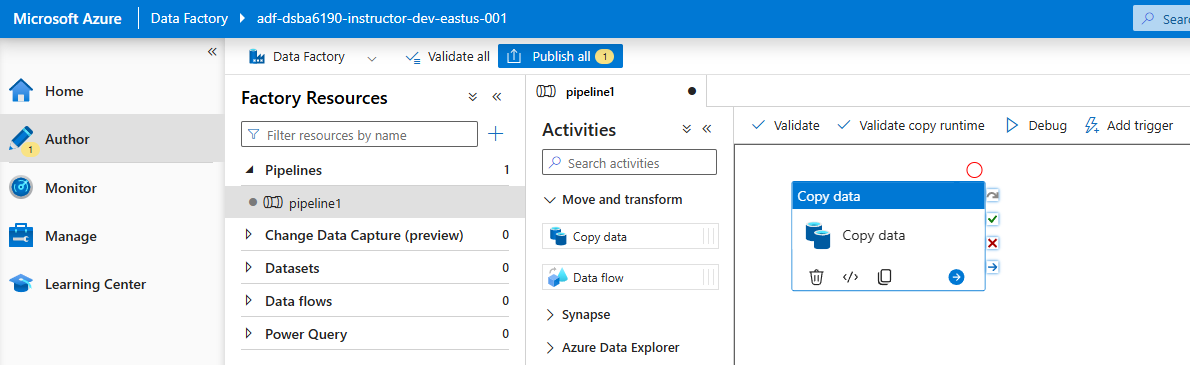
There are a couple ways to copy data using Data Factory.

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. Provide 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 data.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Data Source: Census B03002, an American Census data specifically targeted for Hispanic and Latino Origin.

https://data.census.gov/table/ACSDT1Y2022.B03002

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)?A screenshot of a computer

   Description automatically generated

**$53.05** per person making a total of **$95.65** (for 3 people)

## 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.

Choosing between the two options, a data warehouse would be better if we were working with large amounts of data. They allow for more complex querying and integration from multiple sources. A database is best for transactional systems that handle day to day operations to keep data integrity, not ideal if there are large volumes of data and complex tasks required.

1. What is/are the difference(s) between a star schema and a snowflake schema in a data warehouse?

Star schemas are designed to look like a star with a fact table in the middle and dimension tables going out from it. They are simpler and easy to understand/query since they have fewer joints. They a preferred in environments where simplicity and query are prioritized over storage efficiency. A Snowflake Schema is more normalized but complex where the dimension tables are broken down even further into sub-dimensions. Primarily used in situations where data redundancy needs to be minimized and storage efficiency is more important than query speed.

## Understand platform selection differences.

#### Questions:

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

Azure DB is more for transactional OLTP, better for smaller operations and medium size data volumes, with built in security for a relational structure. Azure Synapse is more for large volumes and complex queries, large scale data processing an integrates a data warehouse and integration into one large scale platform.

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

The cost for Azure DB is around $150 - $1,000 per month, where Synapse can range from $75 - $10,000. This primarily depends on the amount of data you are storing and how complex your query is, or even the amount of people is using it. It might be better for you situation to use Azure DB.

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

Blob is primarily for data storage, flat object based with basic integration, highly scalable for simple use cases, basic security with hot, cool, and archive storage tiers. Data lake is more optimized for big data and analytics, uses hierarchal directories, granular access, deep access to azure analytical tools, used for ML workloads/big data analytics, optimized for big data

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

While both services offer the same tiered pricing for storage (Hot, Cool, Archive) and replication (LRS, ZRS, GRS), data lakes will have slightly higher transaction costs due to its support for hierarchical namespace and more granular access control. If you're just storing large, unstructured objects with few access requirements (e.g., backups, logs), Blob Storage will generally be cheaper. If you're building a complex data lake with lots of hierarchical data and frequent access, ADLS will be more suitable, albeit at a slightly higher cost due to the advanced features.

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

Hierarchical Namespace is the feature that enables the support to create directories and subdirectories, higher workload capability, and improved efficiency of certain operations.

## 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?

A database would be the better option for what they want. It ensures fast query times, supports real-time updates, and provides scalability across different regions. A data warehouse would be more appropriate for reporting, analytics, or historical data aggregation but not for handling real-time transactions in an ordering system.

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

Azure SQL Database is the better option. It is designed to meet the needs of transactional workloads with fast performance and scalability. Azure Synapsewould be better suited for large-scale data analytics and reporting, which could complement the SQL Database later for business intelligence but is not needed for what they want with a real-time ordering system.

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

West Central would be the most ideal of the locations, if that doesn’t work, West Us or West Us 2 would be the next likely options. The latency and performance of will be phenomenal if it is a network in the same region, far away regions will have longer wait times delays. Some other considerations would be Geo-replication, if something was lost via an issue, it would be easier to get a backup at a closer location. Cost is also other reasons as if it’s a location in demand, the price might be higher than other locations.

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.

Database

* Single Database
* Purchase Model: vCore
* Service Tier: General Purpose
* Compute Tier: Provisioned
* Hardware Type: Standard-Series (Gen 5)
* Instance: 4vCore
* Disaster Recovery: Primary or Geo Replica
* Redundancy: Locally Redundant 1 Database for 31 days
* Saving Options: Compute pay as you go, SQL License pay as you go
* All above total = $750.50
* Storage: 500gb x 1 Database x $.115 per gb/month = $57.50
* Log: 150gb x 1 Database x $.115 per gb/month = $17.25
* Backup Storage Redundancy: RA-GRS
* Point in time Restore: 500gb x $.200 per gb/month = $100.00
* Long Term Retention: 3 Yearly Backups x 500gb x $.050 = $75.00
* **Grand Total Monthly Costs: $1,000.25**

**DATABASE IS THE BETTER OPTION**

Data Warehouse

* Region: West Us
* Performance Tier: Compute Optimized Gen 2
* 500 DWU Blocks 31 days x $7.550 per hour = $5,617.20
* Storage: 1TB (because they don’t allow anything lower than that) x $27.75 per TB
* Geo Redundant Disaster Recovery: 1TB x $.07 per gb/month = $69.43
* Memory Optimized: small (4vCore/32gb)
* **Grand Total: $7,057.69**