**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:**

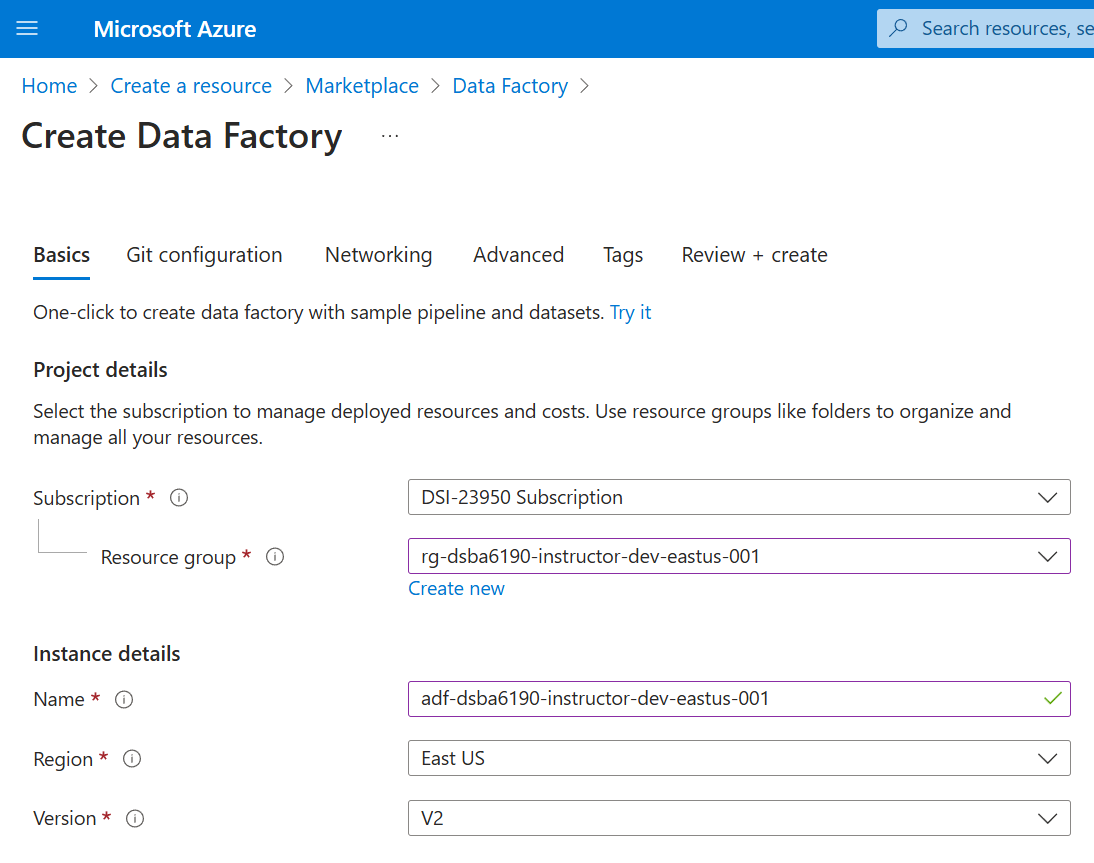
* Only provision the requested resources using the defined settings. Remember, the class cloud budget is everyone’s responsibility.
* Use the standard [Azure naming conventions](https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming) when your name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

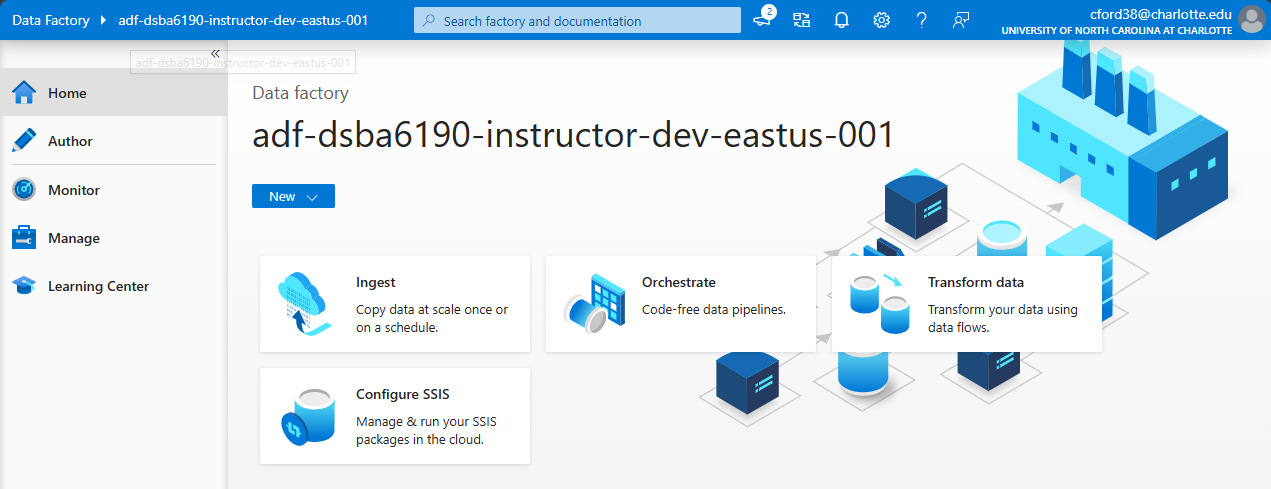
* Documentation: <https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-cli#upload-a-blob>

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>

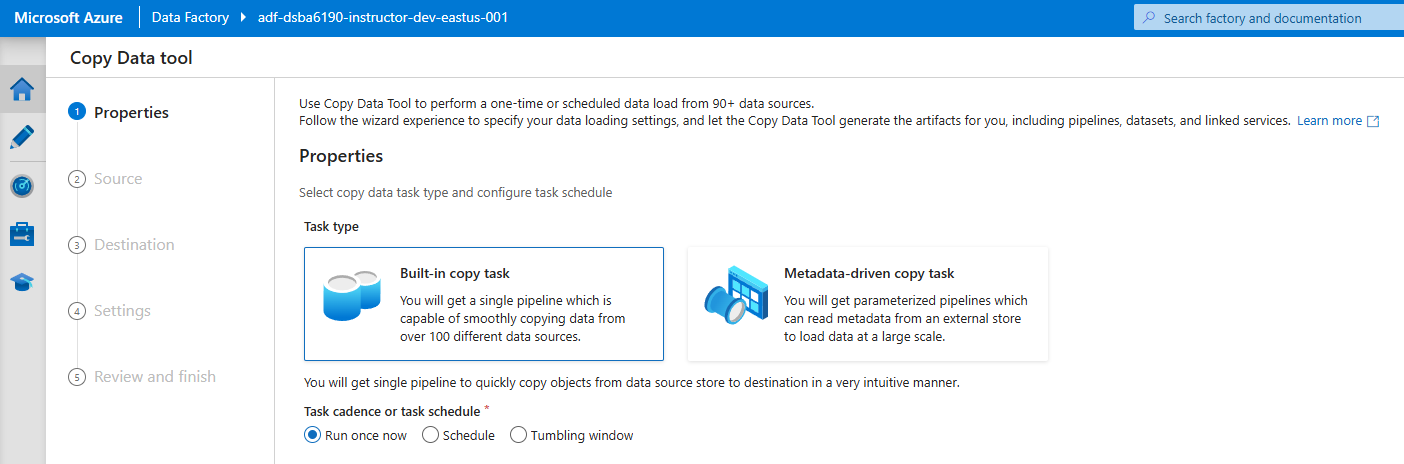
# Step 6: Copy a File using Azure Data Factory

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



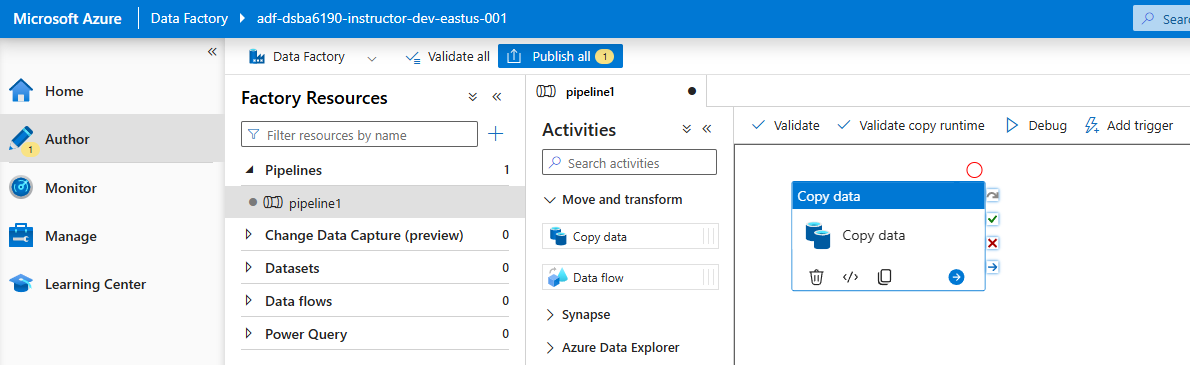
There are a couple ways to copy data using Data Factory.

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.
2. 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.)
3. 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.
4. 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)?

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

## Understand platform selection differences.

#### Questions:

1. What are the capability differences in using Azure DB vs. Azure Synapse?
2. What are the cost differences in using Azure DB vs. Azure Synapse?
3. What are the capability differences in using Azure Blob Storage vs. Azure Data Lake?
4. What are the cost differences in using Azure Blob Storage vs. Azure Data Lake?
5. What was the setting that turned the normal Azure Blob Storage Account into a data lake (Azure 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?
2. Would you recommend University Supplies Corporation use Azure SQL DB or Azure Synapse? Why?
3. Which region would you provision the Azure SQL DB/Synapse? (Bonus: Are there any considerations or capability limitations for choosing this region?)
4. 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.