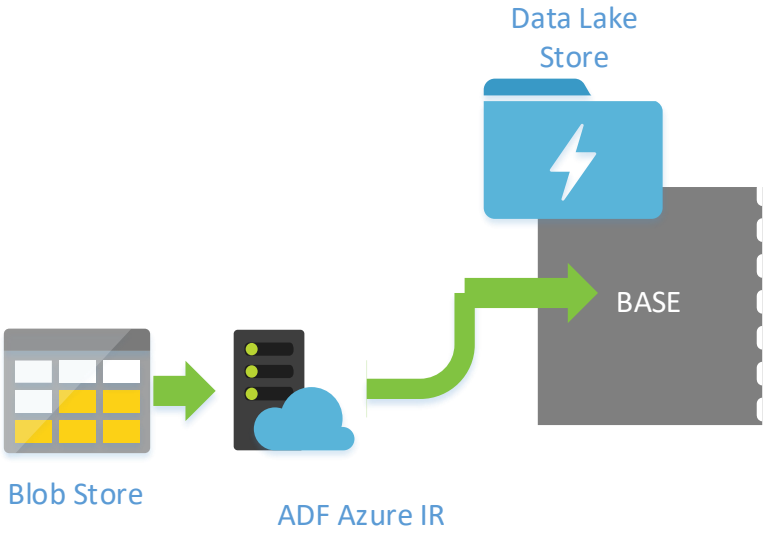
Paul Andrew Simon Whiteley Terry McCann

Lab Overview and Objective

Ingest the first parts of our Adventure Works data source using an Azure Data Factory copy activity from a provided Azure Blob Storage account into an Azure Data Lake Store landing area.

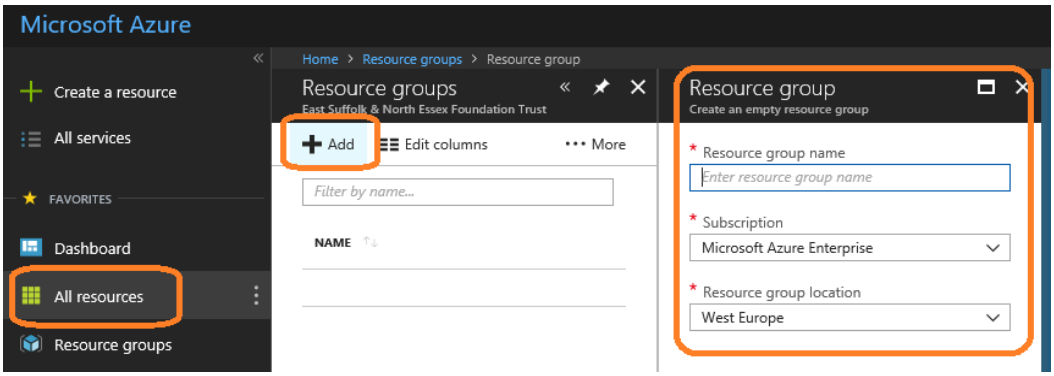
This lab covers the below section of our overall architecture.



Steps

Narrative: firstly, let's get some PaaS tech deployed so we can start developing the data services.

1. Create a Resource Group for this Solution. This is just a convenient contains for our services.



2. Deploy the required Azure services via the Portal.

   a. Create an Azure Data Lake Store (ADLs)
      i. **Type:** Generation 1
      ii. **Region:** West Europe
      iii. **Tier:** Pay-as-you-Go
      iv. **Encryption:** Enabled

b. Create Azure Data Lake Analytics (ADLa)
   i. **Region:** West Europe
   ii. **Storage:** Connect to storage account deployed in part A.
   iii. **Tier:** Pay-as-you-Go

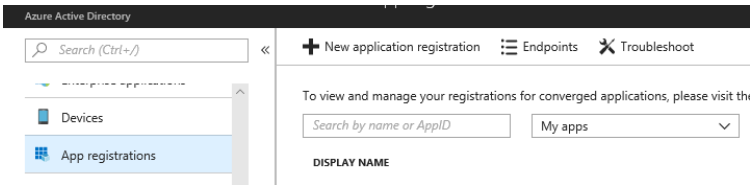
c. Create an Azure Data Factory (ADF)
   i. **Region:** West Europe
   ii. **Version:** 2

Narrative: next, for some services to authenticate with each other a service principal is required. Like an on premises service account. It's easier to create this now rather than later.

3. Create an Azure Service Principal and grant access to Azure Data Lake.

   a. Using the **Azure Portal**, navigate to **Azure Active Directory (AAD)**.

   b. Within AAD go to **App Registrations**.



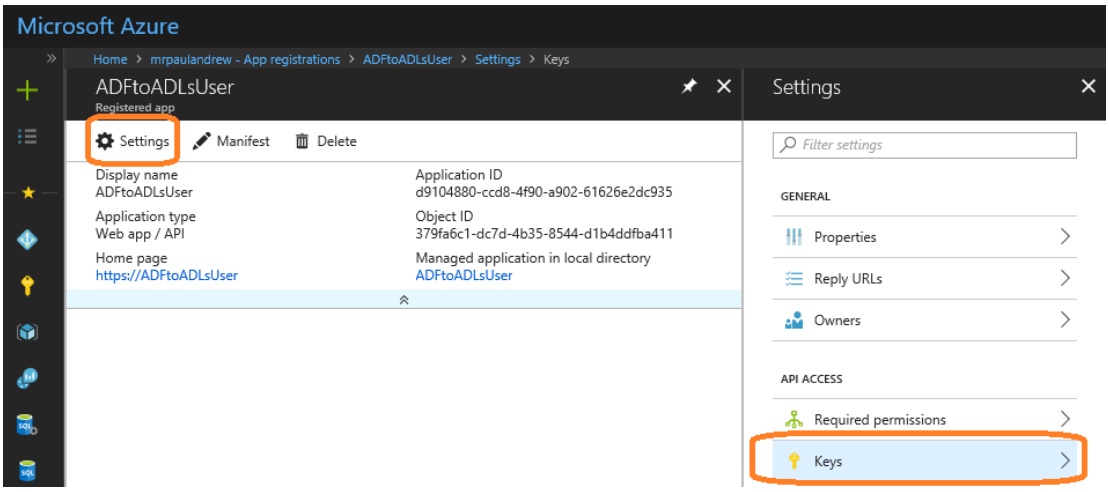
   c. Click **New application registration**. Enter the details below, or some other name if you prefer.

Name	ADFtoADLsUser
Type	Web app/API
Sign-on URL	https://ADFtoADLsUser

   d. Once entered. Click **Create**.

      Copy the **Application ID** value. Save it in a file and location known to you. We'll need this later in Data Factory.

e. In the resulting portal blade from creating the service principal, click **Settings**. Then **Keys**.

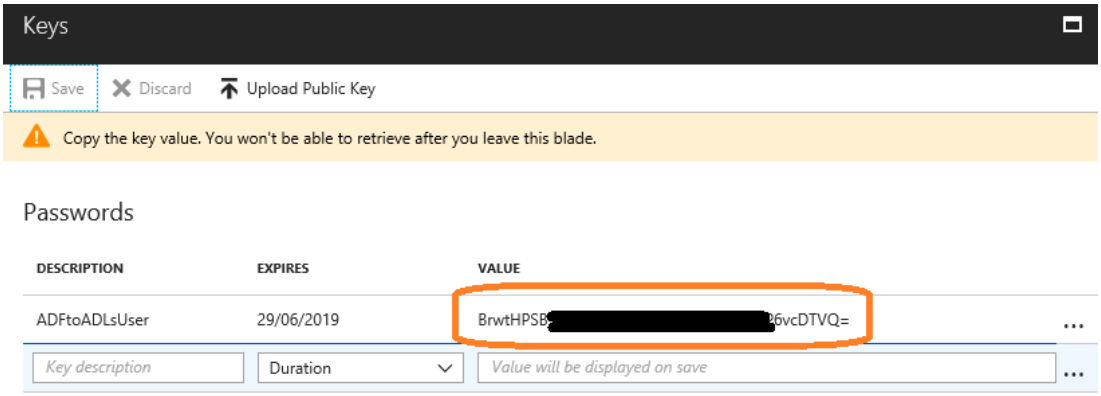


f. In the **Passwords** section of the **Keys** blade enter:

Description	ADFtoADLsUser
Expires	In 1 Year
Value	ADFtoADLsUser

g. Click **Save**.

Copy the **Secret** key value provided. Save it in a file and location known to you. We'll need this later in Data Factory.

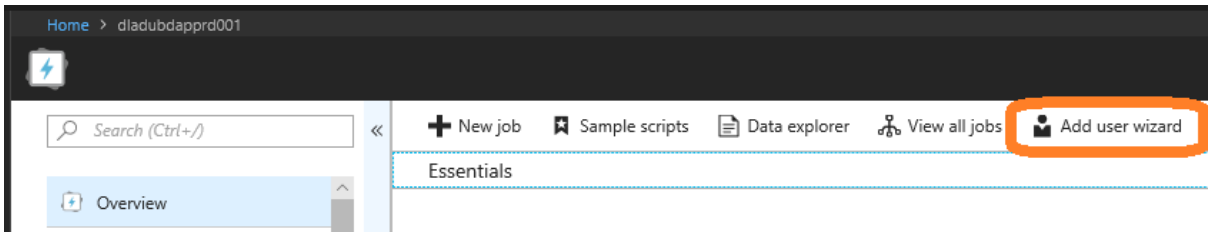


h. You can now close the service principal blades.

Narrative: having created the service principal we now need to grant it access to our Azure Data Lake services. The simplest way to do this is through the ADLa service itself.

i. Using the **Azure Portal**, navigate to your **Azure Data Lake Analytics (ADLa)** service deployed in step 2b.

j. Choose **Add User Wizard** from the top toolbar.



k. In part **1** of the wizard blade enter the name of your service principal 'ADFtoADLsUser'. Or the value you choose.

l. Select the user from the list below and click **Select**.

m. In part **2** of the wizard choose **Owner**. For ease. In production tighter security setting maybe required.

n. In part **3** of the wizard ensure the user has **Read and Write** access to all service components. Then click **Select**.

o. In part **4** of the wizard ensure the user has **Read, Write and Execute** permissions to all folder levels within the storage account. Change 'Apply To' **This folder and all children**. Then click **Select**.

p. In part **5** of the wizard click **Run**.

q. Wait for everything to complete and then click **Done**.

Narrative: we now have an Azure Service Principal with access to our Azure Data Lake services.

Paul Andrew Simon Whiteley Terry McCann

Narrative: Azure Data Lake Storage out of the box contains only system child folders. We need to create a comprehensive set of parent and child directories as ADLs will be our landing area for all files regardless of source and type.

Adatis has an established way of doing this detailed in the following blog post.

http://blogs.adatis.co.uk/ustoldfield/post/Shaping-The-Lake-Data-Lake-Framework

4. Deploy the Adatis Azure Data Lake Storage framework.

    a. Using the **Azure Portal**, navigate to your **Azure Data Lake Store (ADLs)** service deployed in step 2a.

    b. Choose **File Explorer**.

    c. On the toolbar click **New Folder**. Create the following directories:

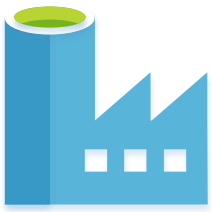
        i. RAW
            1. External
                a. AdventureWorks
                i. Batch
                ii. Streamed
        ii. BASE
        iii. ENRICHED
            1. Warehouse

There is also a PowerScript available in the GitHub repository to create this automatically if you have the required Azure modules installed. Use either VSCode or the PowerShell ISE to do this.

Narrative: now we have a suitable landing area for our data lets copy some into the storage from a potential source system.

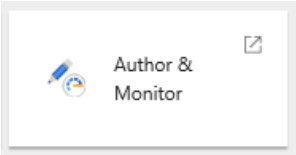
For ease the first source of our data will be an Azure Blob Storage account that we already have setup. We'll copy data from here into your ADLs.

5. Copy data with Azure Data Factory.

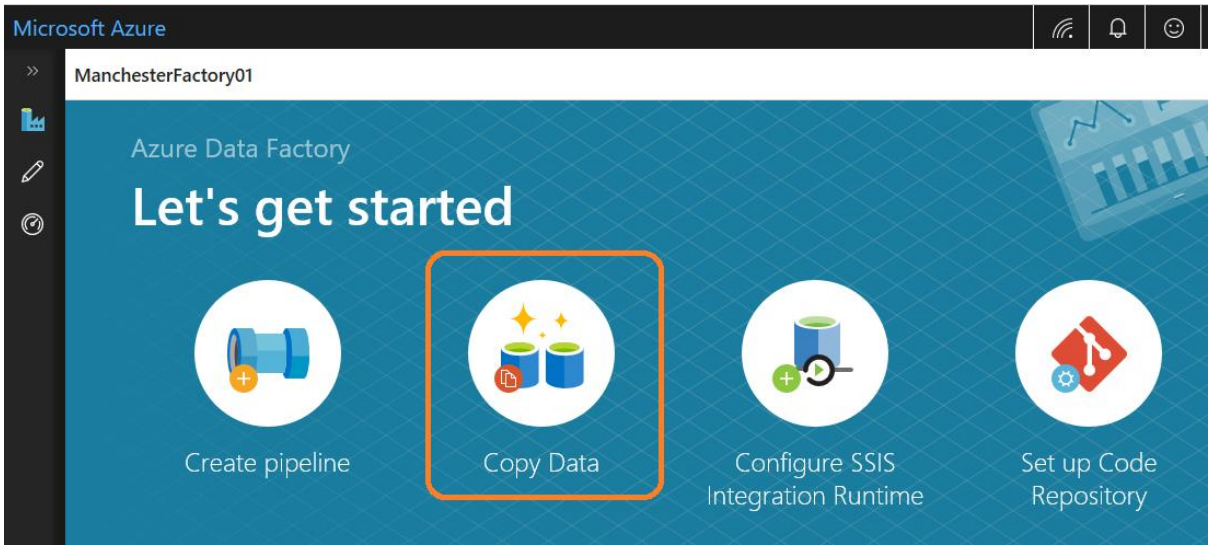


   a. Using the **Azure Portal**, navigate to your **Azure Data Factory v2 (ADF)** service deployed in step 2c.

   b. From the service blade open the **Author and Monitor** tool. This will open as a separate tab in your browser.



   c. Within the ADF sub portal. Under 'Let's get started' choose **Copy Data**.



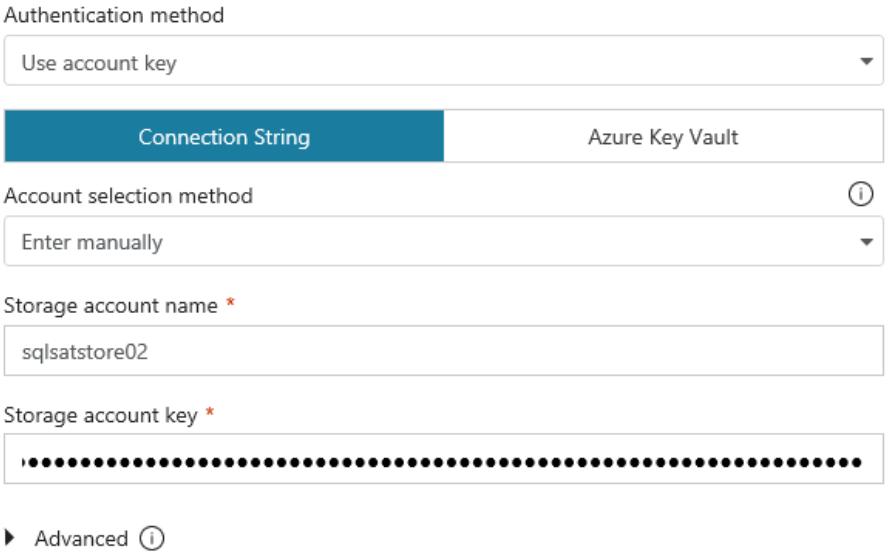
   d. Follow the copy data wizard through to conclusion.

      i. Step 1, give your pipeline a suitable name and choose **Run regularly** with a **Schedule**. Any schedule is fine for now.

Narrative: creating the source ADF linked service and supporting dataset.

      ii. Step 2, click **Create new connection**. Under Azure choose **Azure Blob Storage**. Then click **Continue**.

Complete the connection details manually providing the storage account name and key that can be found in the GitHub repository under lab 1.



iii. Once the connection has been created click **Next**.

iv. Choose **Azure Blob Storage** on the **Connection Properties** panel.

v. Choose the **adventure-works** container.

vi. Select **Copy file recursively**.

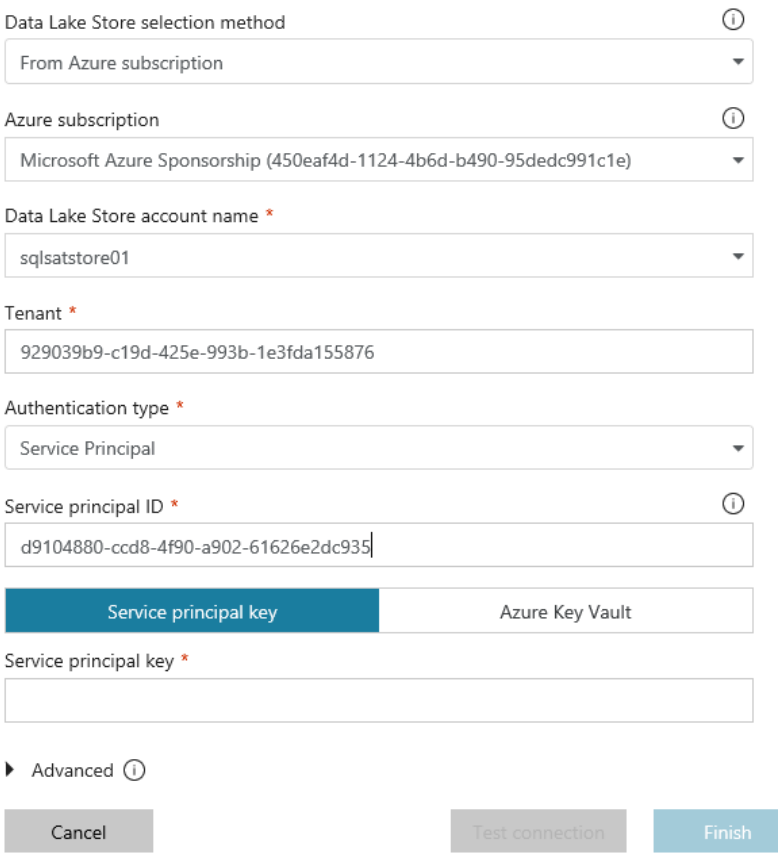
vii. Ignore any file structure and format settings.

Narrative: creating the destination ADF linked service and supporting dataset.

viii. Step 3, click **Create new connection**. Under Azure choose **Azure Data Lake Storage Gen1**. Then click **Continue**.

Complete the connection details choosing the ADLs account created in step 2a.

Set the **Authentication Type** as **Service Principal** and populate the fields with the details of the account created in step 3c.



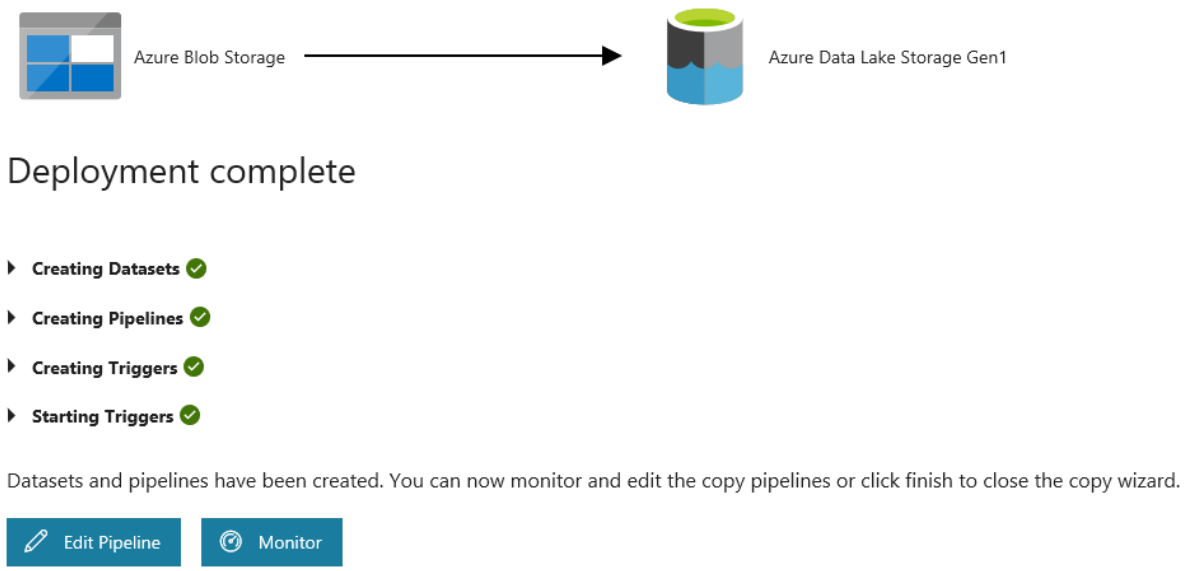
ix. Once entered click **Finish**.

x. Ensure the new ADLs connection is selected and click **Next**.

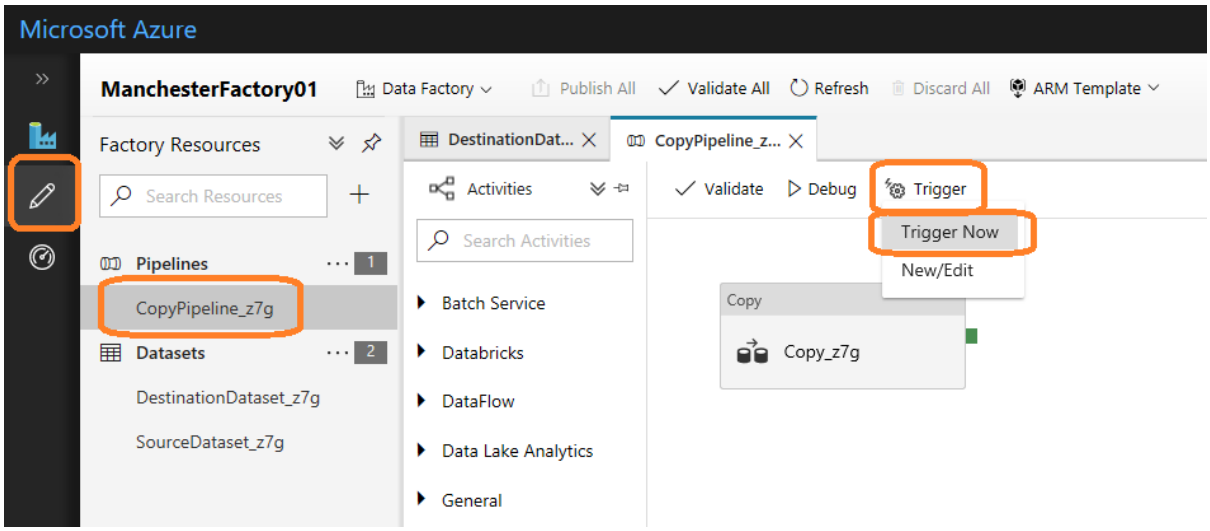
xi. Choose **/RAW/External/AdventureWorks/Batch** as the destination directory.

xii. Ignore the file format setting again for the destination.

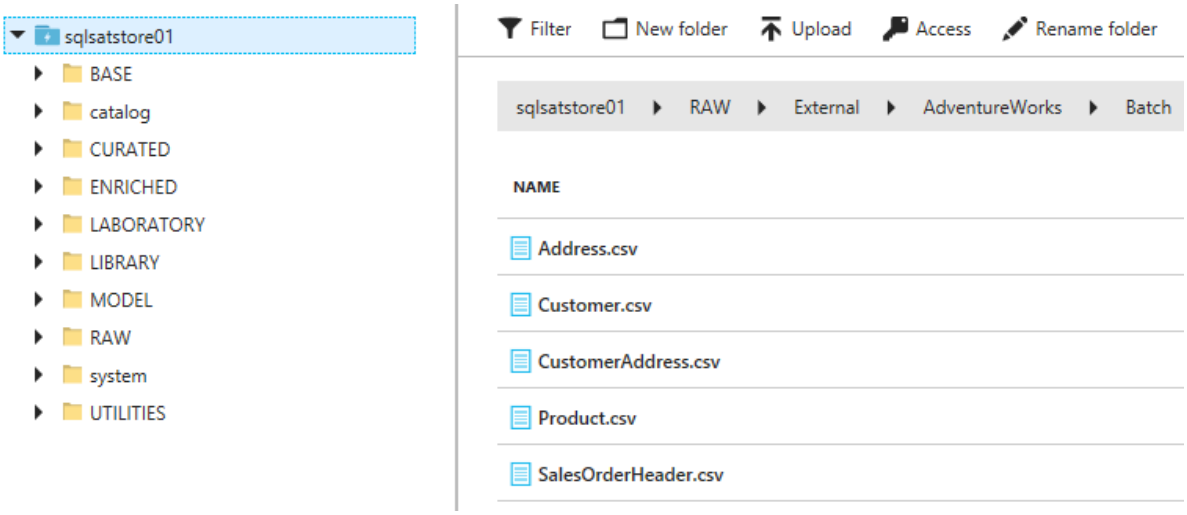
xiii. Review the setting. Allow the pipeline to be created and click **Finish**.



xiv. Return to your data factory. Choose the authoring area. Select your new pipeline and click **Trigger Now** to run the copy.



xv. Navigate to your ADLs account in the main Azure Portal browser tab. Go to Data Explorer and confirm your data has been copied.



That concludes lab 1.

To recap, we've created.

- An Azure Data Lake Storage account.
- An Azure Data Lake Analytics service.
- An Azure Data Factory.
- A service principal with permissions to access our data lake.
- Within ADF we've created:
  - Linked services to source and destination services.
  - Dataset to support the files being copied.
  - A pipeline with a scheduled trigger.
  - A Copy activity to ingest the data into our ADLs.
- We've executed the pipeline and ingested our first datasets.